



João Faro Vale

Licenciado em Ciências da Engenharia Eletrotécnica e de  
Computadores

## Estudo de Moduladores Sigma-Delta Incrementais

MESTRADO EM Engenharia Eletrotécnica e de Computadores  
Universidade NOVA de Lisboa  
Novembro, 2021



DEPARTAMENTO DE Engenharia Eletrotécnica  
e de Computadores

---

## Estudo de Moduladores Sigma-Delta Incrementais

**João Faro Vale**

Licenciado em Ciências da Engenharia Eletrotécnica e de  
Computadores

**Orientador:** Nuno Filipe Silva Veríssimo Paulino  
Professor Doutor, Universidade NOVA de Lisboa

**Júri:**

**Presidente:**

**Arguentes:**

**Orientador:** Nuno Filipe Silva Veríssimo Paulino  
Professor Doutor, Universidade NOVA de Lisboa

**Membros:**

MESTRADO EM Engenharia Eletrotécnica e de Computadores

Universidade NOVA de Lisboa  
Novembro, 2021

Copyright © João Faro Vale, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.



## AGRADECIMENTOS

Quero começar por agradecer não só à faculdade como ao professor Nuno Paulino por toda a atenção e dedicação prestada. Irei levar tudo o que me ensinou para o futuro.

Quero agradecer a todos os meus amigos mais próximos por todo o apoio dado durante estes anos de curso e boas memórias que certamente levarei comigo para a vida.

Gostaria também de agradecer à minha namorada Marta Neves por toda a paciência e apoio.

Por fim, agradeço a toda a minha família, em especial aos meus pais e irmã, por todo o apoio que me deram durante o meu percurso académico, proporcionando-me os melhores tempos de sempre.

A todos vós, Muito Obrigado!



## RESUMO

Esta dissertação consiste inicialmente na realização de uma revisão geral a vários tipos de Moduladores Sigma-Delta implementados de diferentes formas e utilizando várias estratégias diferentes que possibilitam a sua decimação. Foi também realizada a análise mais detalhada ao modulador Sigma-Delta Incremental, tendo sido implementado um modulador de primeira ordem e um modulador de Segunda ordem que foram filtrados com o filtro Col (Cascata de Integradores) e um filtro que se baseia num algoritmo de decodificação ótimo.

Para as arquiteturas que foram desenvolvidas e analisadas detalhadamente, foi observado em simulações no *software Octave* os valores das métricas SNDR e DR de modo a otimizar as arquiteturas. Com esta análise concluiu-se que, para aplicação em sistemas reais, tanto para os moduladores primeira ordem como para os moduladores de segunda ordem, nenhum dos casos analisados (com filtro Col e com filtro ótimo) tem uma grande vantagem perante o outro, ainda assim o filtro ótimo obtém valores ligeiramente superiores.

Palavras-chave: Modulador Sigma-Delta Incremental, Conversor analógico para Digital, Filtro Col, Filtro ótimo, tempo discreto, Relação Sinal-Ruído e Distorção~



## ABSTRACT

This dissertation initially consists of a general review of several types of Sigma-Delta Modulators implemented in different ways and using several different strategies that enable their decimation. A more detailed analysis of the Sigma-Delta Incremental modulator was also carried out, having implemented a first-order modulator and a second-order modulator which were filtered with the Col filter (Cascade of Integrators) and a filter based on an algorithm optimal decoding.

For the architectures that were developed and analyzed in detail, the values of the SNDR and DR metrics were observed in simulations in Octave software, to optimize the architectures. With this analysis it was concluded that, for application in real systems, both for the first order modulators and for the second order modulators, none of the analyzed cases (with Col filter and with optimal filter) has a great advantage over the other, the optimal filter still gets slightly higher values.

Key words: Sigma-Delta Incremental Modulator, Analog to Digital Converter, Col Filter, Optimal Filter, Discrete Time, Signal-to-Noise Ratio and Distortion



# CONTEÚDO

AGRADECIMENTOS .....	III
RESUMO.....	V
ABSTRACT .....	VII
CONTEÚDO.....	IX
LISTA DE FIGURAS.....	XI
LISTA DE TABELAS.....	XV
ABREVIATURAS.....	XVII
<b>1 INTRODUÇÃO .....</b>	<b>1</b>
1.1. Objetivos.....	2
1.2. Motivações.....	2
1.3. Estrutura do documento.....	3
<b>2 ESTADO DA ARTE .....</b>	<b>5</b>
2.1. Conversor analógico digital .....	5
2.2. Métricas para análise da performance .....	6
2.2.1. SNR.....	6
2.2.2. <i>Dynamic range</i> .....	6
2.2.3. SFDR.....	6
2.2.4. SNDR.....	7
2.2.5. ENOB.....	7
2.2.6. Figure of Merit.....	7
2.3. Tipos de ADC .....	8
2.3.1. Flash ADC .....	8

2.3.2.	Pipeline ADC .....	10
2.3.3.	SAR ADC .....	12
2.3.4.	Sigma-Delta.....	13
2.3.4.1.	Filtro Decimador .....	14
2.3.5.	Comparação entre tipos de ADC .....	14
2.4.	ADC Sigma-Delta.....	16
2.5.	ADC Sigma-Delta Incremental .....	19
2.6.	Arquiteturas ADC Sigma-Delta .....	19
2.7.	Arquiteturas ADC Sigma-Delta com decodificação Ótima.....	32
2.8.	Tabela de comparação de resultados.....	37
<b>3</b>	<b>SIGMA-DELTA PRIMEIRA ORDEM.....</b>	<b>39</b>
3.1.	Análise Teórica .....	39
3.2.	Funcionamento do Modulador de 1ª Ordem e Filtros.....	43
3.2.1.	Implementação com Filtro Col.....	45
3.2.2.	Implementação com Filtro Ótimo.....	48
3.3.	Análise de alto nível .....	51
3.3.1.	Ambiente de simulação.....	51
3.3.2.	Simulações de alto-nível .....	54
<b>4</b>	<b>SIGMA-DELTA SEGUNDA ORDEM.....</b>	<b>63</b>
4.1.	Análise Teórica .....	63
4.2.	Funcionamento Modulador de 2ª Ordem e Filtros .....	65
4.2.1.	Implementação com Filtro Col.....	67
4.2.2.	Implementação com Filtro Ótimo.....	69
4.3.	Simulações de alto nível .....	70
<b>5</b>	<b>CONCLUSÃO E TRABALHO FUTURO .....</b>	<b>83</b>
5.1.	Conclusão.....	83
5.2.	Trabalho futuro .....	84
	<b>REFERÊNCIAS.....</b>	<b>85</b>

## LISTA DE FIGURAS

Figura 2.1 - Diagrama de Blocos de um ADC ideal .....	5
Figura 2.2 - Arquitetura de um Flash ADC. ....	8
Figura 2.3 - Codificador Wallace Tree de 3 bits. ....	10
Figura 2.4 - (a) Arquitetura de um <i>pipeline</i> ADC, (b) Exemplo de uma das etapas do <i>pipeline</i> ADC.....	11
Figura 2.5 - Arquitetura de um SAR ADC. ....	12
Figura 2.6 - Arquitetura de um ADC Sigma-Delta. ....	13
Figura 2.7 - Diagrama de blocos do filtro decimador .....	14
Figura 2.8 - Resposta de um filtro decimador (azul).....	14
Figura 2.9 - Amplificador utilizado no primeiro integrador. ....	17
Figura 2.10 - Compensação do excesso de atraso, através do condensador Cx. ....	17
Figura 2.11 - Circuito do DAC utilizado.....	18
Figura 2.12 - Diagrama de blocos (a) Sigma-Delta em tempo discreto, (b) Sigma-Delta em tempo contínuo.....	18
Figura 2.13 - Diagrama de blocos de um Sigma-Delta Incremental.....	19
Figura 2.14 – Diagrama de blocos da arquitetura proposta .....	20
Figura 2.15 - Diagrama de blocos do modulador Sigma-Delta de 3ª ordem.. ....	21
Figura 2.16 - Arquitetura proposta para o modulador Sigma-Delta de 3ª ordem.....	22
Figura 2.17 - Modulador Sigma-Delta de 2 canais com filtro $\pi$ -shifted.....	23
Figura 2.18 - Diagrama de blocos do modulador com o filtro decimador Col. ....	23
Figura 2.19 - Circuito de uma das 4 partes do primeiro integrador.. ....	24
Figura 2.20 - Diagrama de blocos do conversor incremental proposto.....	25
Figura 2.21 - Arquitetura do bloco de controlo.....	25
Figura 2.22 - Diagrama de blocos do modulador Sigma-Delta de 2 andares com filtro digital.....	26
Figura 2.23 - Modulador Sigma-Delta de tempo contínuo de 2 andares .....	27

Figura 2.24 - Arquitetura do Feedforward coefficient summation com quantizador de 1 bit.....	28
Figura 2.25 - a) Diagrama de blocos da arquitetura; b) Versão modelada da mesma arquitetura.....	29
Figura 2.26 - Arquitetura Ota utilizada no primeiro integrador .....	30
Figura 2.27 - Diagrama de blocos do modulador de 2ª ordem com o filtro decimador	31
Figura 2.28 - Correspondência entre a terminologia utilizada em <i>Design</i> de circuitos e a implementada em [28]. .....	34
Figura 2.29 - Directed acyclic graph para o modulador Sigma-Delta de segunda ordem .....	36
Figura 3.1 - Diagrama de blocos do modulador Sigma-Delta de Primeira ordem, para análise teórica .....	39
Figura 3.2 - Diagrama do integrador <i>Switched Capacitor</i> .....	40
Figura 3.3 - a) Circuito correspondente à $\phi 1$ ; b) Circuito correspondente à $\phi 2$ .....	40
Figura 3.4 - Diagrama de blocos do modulador Sigma-Delta Incremental de Primeira ordem.....	43
Figura 3.5 – Integrador Switched Capacitor com realimentação .....	44
Figura 3.6 – Gráfico de comparação da média do sinal de entrada e do sinal de saída em função da amplitude de entrada, $OSR = 1200$ , $N_{medias} = OSR$ .....	45
Figura 3.7 – Filtro Sinc.....	45
Figura 3.8 - Integrador e diferenciador do Filtro Digital .....	46
Figura 3.9 - Filtro com o diferenciador simplificado .....	47
Figura 3.10 - Diagrama de blocos do modulador de primeira ordem implementado com filtro Col de segunda ordem .....	47
Figura 3.11 - Gráfico da saída do primeiro integrador do filtro Col em função das amostras.....	48
Figura 3.12 - Gráfico da saída do segundo integrador do filtro Col em função das amostras.....	48
Figura 3.13 - Diagrama de blocos do Modulador Sigma-Delta de 1ª ordem implementado com Filtro Ótimo.....	49
Figura 3.14 - Gráfico da saída filtro ótimo em função das amostras.....	50
Figura 3.15 - Sinal de entrada $V_{in}$ com taxa de amostragem $OSR$ .....	51
Figura 3.16- Sinal de entrada com $OSR_{sinal} = 10$ .....	52
Figura 3.17- FFT do sinal de entrada com $OSR_{sinal} = 10$ .....	52
Figura 3.18 - Sinal de entrada com $OSR_{sinal} = 1200111$ .....	53

Figura 3.19 - FFT do sinal de entrada com $OSR_{sinal} = 1200111$ .....	53
Figura 3.20 - Gráfico do sinal de entrada com pontos amostrados e com influência do ruído térmico destacado.....	54
Figura 3.21 - Diagrama do modulador de primeira ordem com filtro Col de segunda ordem.....	54
Figura 3.22 - Gráfico de SNDR em função da amplitude de entrada, com ponto máximo de SNDR marcado.....	55
Figura 3.23 - Gráfico de SNDR para as saídas Y1 e Y2 do filtro Col com integradores ideais do modulador reais, $\beta=1$ .....	56
Figura 3.24 - Gráfico de SNDR para as saídas Y1 e Y2 do filtro Col com integradores do modulador reais, $\beta=0,99$ .....	56
Figura 3.25 - Gráfico do SNDR para a variação de OSR.....	57
Figura 3.26 - Gráfico do DR para a variação de OSR.....	57
Figura 3.27 - Gráfico do SNDR para a variação de $V_{nt}$ .....	58
Figura 3.28 - Gráfico do DR para a variação de $V_{nt}$ .....	59
Figura 3.29 - Gráfico do SNDR para a variação do coeficiente $\beta$ .....	60
Figura 3.30 - Gráfico do DR para a variação do coeficiente $\beta$ .....	60
Figura 3.31 - FFT dos sinais de saída com integradores ideais, $\beta=1$ .....	61
Figura 3.32 - FFT dos sinais de saída com integradores reais, $\beta=0,99$ .....	62
Figura 4.1 - Diagrama de blocos do modulador Sigma-Delta de Segunda ordem, para análise teórica.....	63
Figura 4.2 - Diagrama de blocos do modulador Sigma-Delta Incremental de Segunda ordem.....	66
Figura 4.3 - Gráfico de comparação da média do sinal de entrada e do sinal de saída em função da amplitude de entrada para o modulador de segunda ordem.....	67
Figura 4.4 - Diagrama de blocos do Modulador Sigma-Delta de 2ª ordem implementado com Filtro Col de 3ª ordem.....	67
Figura 4.5 - Gráfico da saída do primeiro integrador do filtro Col em função das amostras.....	68
Figura 4.6 - Gráfico da saída do segundo integrador do filtro Col em função das amostras.....	68
Figura 4.7 - Gráfico da saída do terceiro integrador do filtro Col em função das amostras.....	69
Figura 4.8 - Diagrama de blocos do Modulador Sigma-Delta de 2ª ordem implementado com Filtro Ótimo.....	69

Figura 4.9 - Gráfico referente ao sinal de saída do filtro ótimo.....	70
Figura 4.10 - Modulador Sigma-Delta e filtro Col de terceira ordem .....	71
Figura 4.11 - Gráfico de SNDR em função da amplitude de entrada, com ponto máximo de SNDR marcado .....	71
Figura 4.12 Gráfico do sinal de entrada com pontos amostrados e com influência do ruído térmico destacado.....	72
Figura 4.13 - Espectro relativo ao sinal de entrada, $V_{in}$ .....	72
Figura 4.14 - Gráfico de SNDR para as saídas do filtro Col com integradores ideais; $\beta=1$ .....	73
Figura 4.15 - Gráfico de SNDR para as saídas do filtro Col com integradores reais, $\beta=0,99$ .....	74
Figura 4.16 - Gráfico do SNDR para a variação de OSR .....	75
Figura 4.17 - Gráfico de DR para a variação de OSR.....	75
Figura 4.18 - Gráfico do SNDR para a variação de $V_{nt}$ .....	76
Figura 4.19 - Gráfico do DR para a variação de $V_{nt}$ .....	76
Figura 4.20 - Gráfico do SNDR para a variação coeficiente $b_2$ .....	77
Figura 4.21 - Gráfico do DR para a variação coeficiente $b_2$ .....	78
Figura 4.22 - Gráfico do SNDR para a variação do coeficiente $\beta$ .....	79
Figura 4.23 - Gráfico do DR para a variação do coeficiente $\beta$ .....	79
Figura 4.24 - FFT dos sinais de saída com integradores ideais, $\beta=1$ .....	80
Figura 4.25 - FFT dos sinais de saída com integradores reais, $\beta=0,99$ .....	81

## LISTA DE TABELAS

Tabela 2.1 - Tabela de comparação das diferentes topologias de ADC.....	15
Tabela 2.2 - Valores obtidos para as arquiteturas descritas no Estado da Arte .....	38
Tabela 3.1 - Tabela dos parâmetros obtidos através das simulações para modulador de primeira ordem.....	61
Tabela 4.1 - Tabela dos parâmetros obtidos através das simulações para modulador de segunda ordem .....	80



## ABREVIATURAS

AAF	Filtro <i>anti aliasing</i>
ADC	Conversor analógico-digital
CIFF	Cascata de Integradores com <i>feedforward</i>
Col	Cascata de Integradores
DAC	Conversor digital-analógico
DAG	<i>Directed acyclic graph</i>
DR	<i>Dynamic range</i>
FFT	<i>Fast Fourier Transform</i>
FoM	<i>Figure of Merit</i>
IFF	<i>Input Feedforward</i>
MSE	Erro médio quadrático
NTF	Função Transferência do ruído
RP	Fase de Recuperação
S&H	<i>Sample &amp; Hold</i>
SAR	Registo de aproximações sucessivas
SFDR	<i>Spurious free dynamic range</i>
SNDR	Relação Sinal-Ruído e Distorção
SNR	Relação Sinal-Ruído
STF	Função Transferência do Sinal

## INTRODUÇÃO

Um conversor analógico-digital (ADC), converte sinais analógicos em sinais digitais. No mundo atual, esta funcionalidade é muito importante, uma vez que é aplicada numa variedade de dispositivos do quotidiano tais como sistemas de áudio, de vídeo, diversos tipos de sensores de grandezas físicas. Esta funcionalidade permite converter a tensão (ou corrente) proveniente de parâmetros físicos, como luz, pressão, temperatura ou velocidade e converter num sinal binário - sinal digital- tornando-se assim perceptível aos dispositivos digitais. Existem vários tipos de ADC sendo os mais comuns os Flash, *Pipeline*, Registo de aproximações Sucessivas (SAR) e Sigma-Delta ADC, cada um deles tem as suas características, vantagens e desvantagens que serão descritas no estado de arte.

Em muitos sistemas, é necessária a conversão de mais do que um sinal analógico. Assim sendo, a solução encontrada para alguns dos ADC é a multiplexagem à entrada dos sinais analógicos e, após a conversão o sinal digital é desmultiplexado. Relativamente aos conversores Sigma-Delta esta técnica não é factível, devido ao seu filtro decimador. Estes filtros têm uma resposta impulsiva demasiado longa, uma vez que o sinal digital na saída não tem uma correspondência temporal exata com o sinal de entrada. Para ultrapassar este problema é possível inicializar a memória do modulador Sigma-Delta e do filtro decimador antes de cada conversão. Existem várias arquiteturas testadas para conversão sem memória utilizando os conversores Sigma-Delta, sendo algumas descritas no estado de arte.

O ADC Sigma-Delta incremental é baseado num Sigma-Delta ADC comum, porém o modulador é reinicializado após  $N$  ciclos de relógio, fazendo com que a memória do modulador seja apagada. Este método de reinicialização possibilita que a conversão seja feita

como num ADC à taxa de *Nyquist*. Estes conversores têm sido estudados em tempo contínuo ou em tempo discreto, como apresentado no estado de arte.

## 1.1. Objetivos

O objetivo desta dissertação é num primeiro momento fazer uma revisão geral de vários tipos de Moduladores Sigma-Delta e diferentes formas de implementar o respetivo filtro decimador. O segundo ponto, mais específico, é realizar a análise pormenorizada ao modulador Sigma-Delta Incremental, apresentando algumas soluções para filtrar o seu sinal de saída. Feita a análise os circuitos serão implementados em alto nível com um modulador de primeira ordem e um modulador de segunda ordem, sendo cada um deles filtrado por um filtro cascata de integradores (Col) e um filtro que se sustenta num algoritmo ótimo de decodificação. Com esta modulação será possível avaliar as performances das arquiteturas através de métricas, com a análise da relação sinal-ruído e distorção existente nos sinais de saída. Estas métricas vão ajudar a definir os parâmetros ideais de modo a conseguir dimensionar uma arquitetura otimizada principalmente a nível de relação sinal-ruído e distorção (SNDR).

## 1.2. Motivações

Atualmente, devido à evolução a nível tecnológico, a conversão analógica para digital aumentou consideravelmente a sua relevância no que diz respeito a sensores. O funcionamento desta tecnologia tem por base a receção de sinais analógicos provenientes do mundo real, e a sua conversão para sinais digitais, de modo a puderem ser lidos por dispositivos tecnológicos. Esta transformação é feita recorrendo à utilização de conversores, sendo os Sigma delta mais frequentemente empregues para sensores de temperatura, sensores de imagem e sensores de pressão.

As áreas onde se têm destacado cada vez mais estes conversores é na biomedicina e em telecomunicações. Na área da biomedicina tem sido a utilização de sensores de temperatura que tem ganho muito relevo, já na área das telecomunicações o principal destaque vai para os aparelhos eletrónicos, onde este tipo de conversor ganhou mais importância.

### 1.3. Estrutura do documento

Este documento é constituído pelo estado de arte, onde inicialmente é feita uma breve introdução sobre o ADC. De seguida é feita uma breve descrição e sugeridas expressões sobre as várias métricas que podem caracterizar um conversor. Posto isto é feita uma breve análise aos vários tipos de ADC, que será concluída com uma tabela de comparação dos seus comportamentos. Ainda no estado de arte são descritos mais detalhadamente conversores Sigma-delta de tempo contínuo ou discreto e por fim é realizada uma análise sobre os Sigma-Delta Incremental e comparados os vários parâmetros que cada arquitetura possui e ADC existentes.

Posteriormente a dissertação divide-se em dois capítulos, sendo o primeiro dedicado ao Modulador Sigma-Delta de Incremental de 1ª ordem e o segundo dedicado ao Modulador Sigma-Delta de Incremental de 2ª ordem. Em ambos os capítulos é abordada a análise teórica, o modo de funcionamento do modulador e de cada um dos filtros. Em último lugar são apresentadas as simulações de alto nível realizadas no *Software Octave*.



## ESTADO DA ARTE

### 2.1. Conversor analógico digital

Os ADC convertem sinais analógicos em sinais digitais. Neste caso, o sinal de entrada é analógico contínuo no tempo e de amplitude variável e o sinal obtido à saída é digital discreto no tempo. Esta conversão é feita através da representação dos valores analógicos do sinal de entrada num número limitado de códigos digitais à saída, sendo que cada código representa uma fração da gama analógica de entrada.

Um ADC ideal, representado na Figura 2.1, consiste num sinal de entrada contínuo em tempo e amplitude -  $V_{in}$  - que é amostrado num dispositivo interno de amostragem Sample & hold (S&H), processo de amostragem e retenção. De seguida este é quantizado em amplitude pelo quantizador e assim, à saída do quantizador, é obtido o sinal de saída amostrado e quantizado.



Figura 2.1 - Diagrama de Blocos de um ADC ideal

No S&H o sinal é amostrado intervaladamente com determinada frequência de amostragem -  $F_s$  - ficando retido no intervalo de tempo entre amostragens. Esta amostragem é realizada num certo instante de tempo -  $kT_s$  - sendo  $T_s$  o período de amostragem e  $k$  o número de amostra. O sinal de saída não é alterado no intervalo de tempo em que é amostrado, mas sim no intervalo de tempo da amostra seguinte. Isto é, o valor da amostra é retido até que haja uma nova amostra. Neste processo, é possível a perda de

informação se a frequência do sinal for superior a metade da frequência de amostragem. (teorema de *Nyquist*).

## 2.2. Métricas para análise da performance

Para avaliar a performance de um conversor existem varias metricas que devem ser consideradas. Nos próximos pontos esta contemplada uma breve descrição de cada um desses pontos

### 2.2.1. SNR

A relação sinal ruído (SNR) define a razão entre a potência do sinal e a potência do ruído, sendo que, um valor superior de SNR equivale a um efeito de ruído menor no sinal. A expressão (2.1), representa o cálculo de SNR em dB.

$$SNR = 10 \log_{10} \frac{P_{sinal}}{P_{ruído}} \quad (2.1)$$

### 2.2.2. *Dynamic range*

O *dynamic range* (DR) consiste na relação entre a amplitude do sinal quando o SNR é máximo,  $A_{in_{maz}}$ , e a amplitude do sinal para SNR=0,  $A_{in_0}$ , como apresentado na expressão (2.2).

$$DR = 20 \log_{10} \frac{A_{in_{maz}}}{A_{in_0}} \quad (2.2)$$

### 2.2.3. SFDR

O *spurious free dynamic range* (SFDR) surge devido ao ruído e à não linearidade dos componentes do conversor, sendo importante para saber o valor mínimo de sinal que está livre de distorção [1]. Este valor é calculado, como apresentado na expressão (2.3), pela diferença entre o valor máximo do sinal de entrada,  $V_{in}$  e o valor mais alto de distorção.

$$SFDR = Valor\ máximo\ de\ V_{in} - Valor\ máximo\ de\ distorção \quad (2.3)$$

## 2.2.4. SNDR

A relação sinal-ruído e distorção (SNDR) também conhecido como SINAD, é bastante importante na avaliação da performance de um ADC, uma vez que envolve todos os componentes que provocam ruído ou distorção. O SNDR é calculado pela expressão (2.4), que representa a divisão entre a soma das potências do sinal, do ruído e da distorção e a soma da potência de ruído e distorção, em dB.

$$SNDR = 10 \log_{10} P_{sinal} - 10 \log_{10} (P_{sinal} + P_{ruído}) \quad (2.4)$$

## 2.2.5. ENOB

O *effective-number-of-bits* - ENOB - é obtido recorrendo à expressão (2.5) que representa o SNR para um ADC de  $N$ -bits ideal [2], sendo  $N$  o número de bits.

$$SNR = 6,02 * N + 1,76 \quad (2.5)$$

Resolvendo a equação (2.5) em ordem ao número de bits, que será o ENOB, e substituindo o SNR pelo SNDR obtém-se a expressão (2.6), para o valor de ENOB.

$$ENOB = \frac{SNDR - 1,76}{6,02} \quad (2.6)$$

## 2.2.6. Figure of Merit

A figura de mérito, *Figure of Merit* (FoM), pode ser obtido através de duas expressões diferentes propostas [3], sendo que cada uma tem significados díspares.

$$FoM_W = \frac{P}{2^{ENOB} * F_N} [fj / conv - steps] \quad (2.7)$$

Tendo em conta a expressão (2.7), onde  $P$  é a potência no ADC e  $F_N$  a frequência de *Nyquist*, quanto maior for o valor de  $FoM_W$  menor será a eficiência do ADC. A outra abordagem para o  $FoM_s$  é calculada pela expressão 2.8.

$$FoM_W = DR + 10 \log_{10} \frac{B}{P} [dB] \quad (2.8)$$

Na expressão (2.8), LB a largura de banda e P a potência, quanto maior for o  $FoM_s$  maior a eficiência do ADC.

## 2.3. Tipos de ADC

Existem vários tipos de ADC sendo que cada um tem um funcionamento, características e aplicações diferentes. Desses destacam-se os Flash ADC, os SAR ADC, os *pipeline* ADC e Sigma-Delta ADC. Assim sendo, os pontos seguintes irão ter como objetivo a descrição destes tipos de ADC.

### 2.3.1. Flash ADC

Os Flash ADC são constituídos por  $2N-1$  comparadores, em que  $N$  é o número de bits de um codificador. Como se pode observar através da arquitetura representada na Figura 2.2 este ADC utiliza uma conversão paralela com  $2N-1$  comparadores [4].

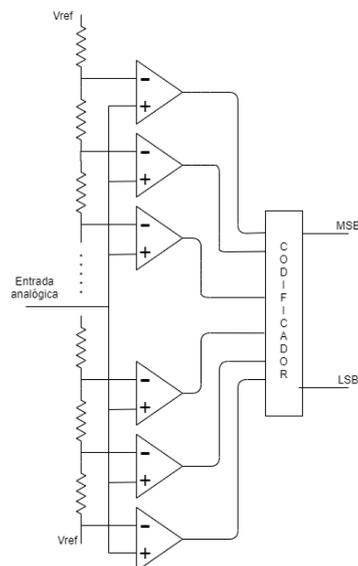


Figura 2.2 - Arquitetura de um Flash ADC. Adaptada de: "Understanding Flash ADCs." [Online]. Available <https://www.maximintegrated.com/en/design/technical-documents/tutorials/8/810.html>. [Accessed: 05-Feb-2020].

O funcionamento de um *flash* ADC é baseado em comparadores onde o sinal de entrada é comparado com uma tensão de referência. Posteriormente, caso o sinal de entrada seja superior à tensão de referência o comparador devolve 1. Se contrariamente tal não

se verificar, o comparador devolve 0, ou seja, os comparadores que estão acima do comparador que tem a entrada analógica ligada devolvem 0. Assim, à saída dos comparadores, o que obtemos é um código termométrico. Por sua vez, os comparadores estão ligados a um codificador primário que é responsável pela conversão do código originando uma saída binária.

Estes conversores são os mais rápidos face às restantes variedades de ADC, sendo também os mais simples, visto que apenas necessitam dos comparadores e da lógica no codificador, são, portanto, os que necessitam de menos componentes. No entanto, o flash ADC apresenta algumas desvantagens, como por exemplo o facto de se tornar bastante mais dispendioso especialmente para mais de 8 bits. Em tal circunstância, será necessário um número elevado de comparadores ( $2N - 1$ ) tornando-se assim uma arquitetura mais apropriada para baixas resoluções. Adicionalmente, são também mais rápidos o que leva a uma dissipação de um valor de potência bastante superior relativamente a outros tipos de ADC, bem como a um aumento na área do chip. Em relação ao sigma-delta não existe comparação pois possuem características bastante distintas. Tendo em conta todas estas características, este tipo de ADC é normalmente utilizado em aplicações que necessitam de baixas resoluções, podendo ser a única arquitetura viável para altas frequências. Assim sendo, pode ser aplicado em recetores de rádio, radares de deteção e sistemas de comunicação ótica.

Uma arquitetura proposta [4] é bastante idêntica à apresentada na Figura 2.3, sendo que esta utiliza um codificador *Wallace tree* que possui baixa potência de dissipação. Este codificador contabiliza a quantidade de '1' que existem no código à saída do comparador e de seguida, os somadores completos que compõem o codificador, representado na Figura 2.3, fornecem o código binário de saída, através da contagem dos números de '1' que têm na sua entrada.

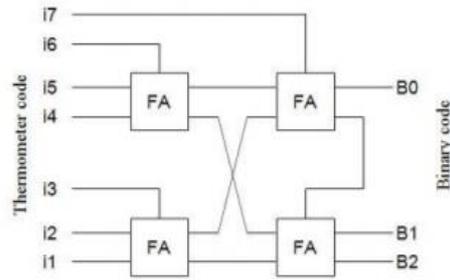


Figura 2.3 - Codificador Wallace Tree de 3 bits. Fonte: M. P. Ajanya and G. T. Varghese, "Low Power Wallace Tree Encoder for Flash ADC," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 396, no. 1, 2018, doi: 10.1088/1757-899X/396/1/012042.

O número de somadores completos necessários é calculado pela expressão (2.9), onde  $N$  representa o número de bits do decodificador.

$$\sum_{i=1}^N (i - 1) * 2^{(N-i)} \quad (2.9)$$

Esta é assim uma forma de diminuir a potência dissipada por este tipo de ADC, reduzindo também os erros no código termométrico.

### 2.3.2. Pipeline ADC

O *pipeline* ADC é constituído por várias etapas sendo cada uma delas constituída por um circuito S&H, um ADC, um Conversor digital-analógico (DAC), um subtrator, um amplificador, um dispositivo de correção digital de erros e um dispositivo de calibração digital (Figura 2.4). O método de conversão utilizado por este ADC é conversão paralela [5].

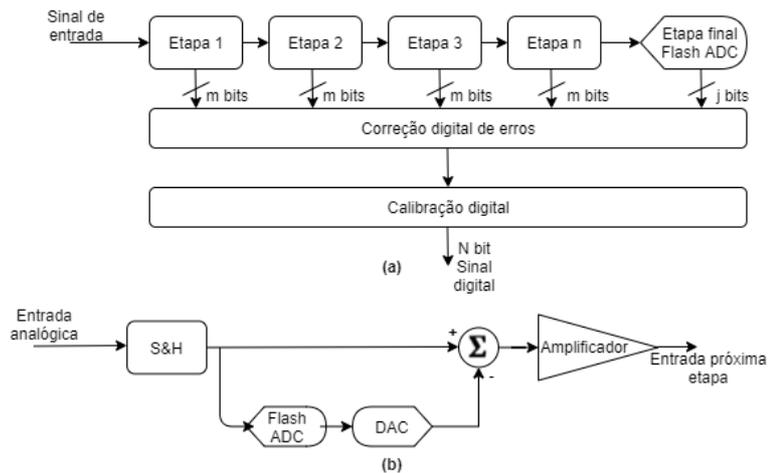


Figura 2.4 - (a) Arquitetura de um *pipeline* ADC, (b) Exemplo de uma das etapas do *pipeline* ADC.  
 Adaptada de: V. Kledrowetz and J. Haze, "Basic block of pipelined ADC design requirements,"  
*Radioengineering*, vol. 20, no. 1, pp. 234–238, 2011.

Inicialmente, o sinal é amostrado pelo circuito S&H ao mesmo tempo que é convertido num ADC que se comporta como um *Flash* ADC de onde vem o sinal digital de saída. Este sinal servirá de entrada a um DAC que converterá o sinal digital em analógico e de seguida será subtraído ao sinal amostrado no S&H, resultando um valor residual de tensão. Esta tensão será amplificada num amplificador de saída, sendo este sinal que vai alimentar a entrada da etapa seguinte. O processo descrito anteriormente é repetido de etapa em etapa até à etapa final que é constituído apenas por um conversor *Flash* ADC. O código de saída pode ser obtido através da soma dos códigos digitais resultantes de cada uma das etapas, que depois passará por uma correção digital de códigos e de seguida por uma calibração digital, que irá garantir maior precisão e desempenho dinâmico.

Um ADC *pipeline* tem como vantagens a flexibilidade em relação à resolução, sendo que esta poderá variar entre os 8 e os 11 bits, sem calibração digital, ou até 18 bits, com calibração digital. Este tipo de ADC tem também uma potência dissipada reduzida. Recentemente o *pipeline* ADC teve alguma evolução em relação à velocidade, resolução e potência dissipada. No entanto, é de referir que as aplicações que necessitam taxas de amostragem mais baixas, usualmente utilizam o SAR ADC ou o Sigma-Delta. Por outro lado, quando se pretende taxas de amostragem mais elevadas a escolha recai por um *Flash* ADC.

Este ADC, tendo em conta as suas resoluções e taxas de amostragem, é utilizado essencialmente em imagens médicas ultrassónicas, recetores digitais e *modems* por cabo.

### 2.3.3. SAR ADC

O SAR ADC é constituído por um S&H, um bloco de lógica SAR, um registo de  $N$  bits, um DAC de  $N$  bits e um comparador [6], como se pode verificar na Figura 2.5.

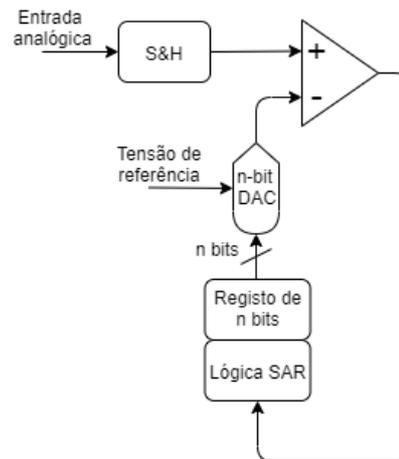


Figura 2.5 - Arquitetura de um SAR ADC. Adaptada de: "Guide to Understanding Successive Approximation Registers (SAR) and Flash ADCs | Maxim Integrated." [Online]. Available: <https://www.maximintegrated.com/en/design/technical-documents/tutorials/1/1080.html>. [Accessed: 05-Feb-2020].

O sinal analógico de entrada é amostrado e retido no S&H enquanto o DAC tem como entrada uma tensão de entrada  $V_{ref}$  e tem como tensão de saída  $V_{ref}/2$ , devido ao registo de  $N$  bits. Após esta conversão o sinal de saída do DAC será comparado com o sinal que foi amostrado. Caso o valor da tensão do sinal amostrado seja superior ao valor da tensão de saída do DAC, a saída do comparador terá o valor lógico 1 e assim o bit mais significativo do registo de  $N$  bits permanece a 1. Caso se verifique a situação contrária, a saída do comparador terá o valor lógico 0 e o bit mais significativo passará a ter o valor 0. Após a conclusão deste processo, o bloco de lógica SAR vai passar ao próximo bit, colocá-lo a 1 e de seguida realizar uma nova comparação, seguindo assim até ao bit menos significativo, quando concluído o código digital estará disponível no registo de  $N$  bits.

Este tipo de ADC tem como principais vantagens a variação entre 8 e 16 bits e o baixo consumo de energia. Em termos de velocidade quando comparado com um *Flash* ADC, o *Flash* será mais rápido, contudo não é viável utilizar um *Flash* ADC de mais de 8 bits,

devido ao número de comparadores necessários. Um SAR ADC é uma boa solução quando se quer obter altas resoluções. Independentemente do inconveniente de serem necessários elementos mais precisos, a complexidade do ADC não irá ter um aumento considerável, pois não é necessário acrescentar um grande número de componentes. Em relação aos Sigma-Delta, os SAR ADC têm a desvantagem de necessitar de calibração, ao contrário do sigma-Delta que não necessita dessa calibração mesmo para altas resoluções, adicionalmente o Sigma-Delta não necessita de filtro *anti-aliasing* (AAF) complexo, uma vez que a taxa de amostragem é superior à largura de banda.

### 2.3.4. Sigma-Delta

Um Sigma-Delta ADC convencional é um conversor de sobre-amostragem que na sua arquitetura [7] tem o integrador, um quantizador (que é um comparador), um DAC e um filtro digital e decimador como mostrado na Figura 2.6.

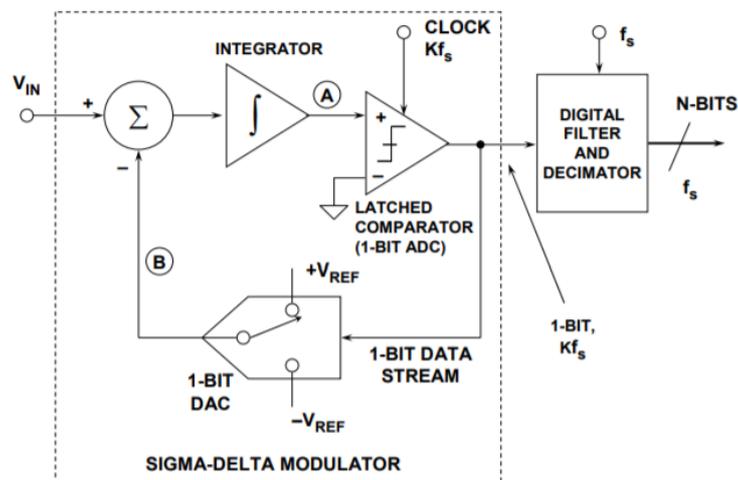


Figura 2.6 - Arquitetura de um ADC Sigma-Delta. Fonte: Y. Lai, "ADC Architectures IV: Sigma-Delta ADC Advanced Concepts and Applications," *Imid 2009*, pp. 1069–1072, 2009.

Inicialmente o sinal é integrado pelo filtro de *loop*, que é constituído por um ou mais integradores dependendo da ordem do modulador. Posteriormente, a gama contínua de amplitudes é quantizada numa gama de níveis discretos. Por fim, é atribuído a cada nível da saída digital um valor binário, pelo filtro digital. O Sigma-Delta pode ser implementado em tempo discreto ou em tempo contínuo, como será descrito de seguida.

### 2.3.4.1. Filtro Decimador

O filtro decimador, que se caracteriza por ter uma resposta muito longa, encontra-se á saída do modulador Sigma-Delta. Este deve ser um filtro passa-baixo, que tem como função filtrar o ruído de quantização fora da banda e amostrar o seu sinal. À posteriori, o ruído será sujeito a um redutor de taxa de amostragem, como representado na Figura 2.7. É ainda de referir que o filtro pode ser implementado de várias maneiras diferentes, sendo que algumas vão ser descritas adiante. Na Figura 2.8 está um exemplo de resposta em frequência de um filtro decimador[8].



Figura 2.7 - Diagrama de blocos do filtro decimador

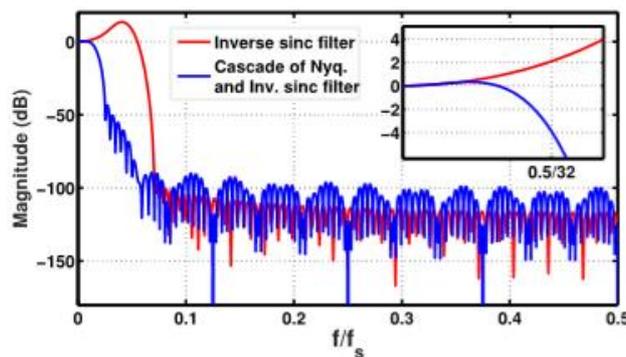


Figura 2.8 - Resposta de um filtro decimador (azul). Fonte: R. S. A. Kumar, D. Behera, and N. Krishnapura, "Reset-Free Memoryless Delta-Sigma Analog-to-Digital Conversion," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 65, no. 11, pp. 3651–3661, 2018, doi: 10.1109/TCSI.2018.2854707.

### 2.3.5. Comparação entre tipos de ADC

De modo a ser possível uma análise mais eficaz dos vários tipos de ADC descritos anteriormente, a informação foi organizada na Tabela 2.1, permitindo retirar algumas conclusões.

Tabela 2.1 - Tabela de comparação das diferentes topologias de ADC

Flash ADC	<i>Pipeline</i> ADC	SAR ADC	Sigma-Delta ADC
Frequência de amostragem alta	Frequência de amostragem média/alta	Frequência de amostragem média	Frequência de amostragem baixa
Área grande	Área grande	Área pequena	Área média
Velocidade alta	Velocidade média	Velocidade baixa	Velocidade baixa
Baixa resolução (4 a 8 bits)	Média resolução (8 a 18 bits)	Média resolução (até 14 bits)	Alta resolução (>14 bits)
Consumo alto	Consumo médio	Consumo baixo	Consumo baixo
Alto custo	Alto custo	Custo médio	Baixo custo

Preliminarmente, é inteligível que todas as arquiteturas têm os seus pontos fortes e fracos. Como tal, é importante saber qual a aplicação designada ao conversor, de modo a optar pela melhor topologia.

A título de exemplo, fazendo uma análise comparativa entre o Sigma-Delta e os restantes tipos de ADC, é possível concluir, que o custo do *Flash* ADC será superior ao de um Sigma-Delta ADC. Esta desigualdade pode ser justificada pela quantidade de componentes necessários para um *Flash* ADC ( $2^{n-1}$  comparadores) atingir a mesma resolução que um Sigma-Delta, tendo ainda a inconveniência de aumentar o tamanho deste conversor. Outro ponto em que o Sigma-Delta também se revela superior ao *Flash* é no que concerne ao consumo de energia, pois com o aumento dos comparadores o consumo vai aumentando. Por outro lado, o *Flash* ADC revela-se melhor face ao Sigma-Delta em termos da velocidade.

Fazendo agora a análise em relação ao *Pipeline* ADC, é possível concluir que operam a uma frequência de amostragem completamente distinta. Estes ADC têm também resoluções relativamente diferentes, visto que no máximo um *pipeline* ADC atinge 16 bits, com calibração digital, sendo o mais próximo da resolução que o Sigma-Delta atinge. Mais uma vez o ponto onde o *Pipeline* ADC se destaca relativamente ao Sigma-Delta é na sua velocidade, onde atinge níveis superiores.

O SAR ADC é o que tem a performance mais semelhante à do Sigma-Delta. Se for necessário ocupar um pequeno espaço e não for necessária alta resolução ou alta

velocidade, o SAR ADC mostra-se a melhor das opções. O ponto a favor do Sigma-Delta em relação ao SAR ADC é o baixo custo.

Concluindo, para uma baixa frequência de amostragem, um baixo consumo, um baixo custo ou uma alta resolução, a melhor opção será um Sigma-Delta ADC.

## 2.4. ADC Sigma-Delta

Os conversores Sigma-Delta podem operar de duas formas diferentes, em tempo discreto ou em tempo contínuo, sendo ambas descritas de seguida:

Sigma-Delta tempo discreto: Num Sigma-Delta ADC de tempo discreto[9], Figura 2.12a, o sinal de entrada é amostrado antes do filtro de *loop*, sendo que tanto estes filtros como o DAC são de tempo discreto. Neste ADC é necessário haver uma transformação de tempo discreto para contínuo para criar o sinal de *feedback*. Um exemplo de uma arquitetura que opera em tempo discreto é proposto em [10], sendo que é uma arquitetura que possui um baixo FoM<sub>w</sub>, que tem como objetivo realizar um quantizador e para reduzir a potência dissipada, realizar compensação do atraso excessivo.

Esta arquitetura utiliza um filtro de *loop* de terceira ordem. O primeiro dos integradores é constituído por um *ampop* que possui uma etapa de entrada PMOS, sendo que os restantes integradores utilizam uma etapa de entrada NMOS. Todos estes integradores utilizam amplificadores operacionais. Na Figura 2.9 está representado o amplificador operacional utilizado no primeiro integrador. Este integrador é responsável pelo ruído e pela linearidade do modulador, utilizando transístores de canais longos de modo a baixar o ruído referido à entrada. Recorre ainda a compensação de *Miller*, representada por  $R_c$  e  $C_c$ .

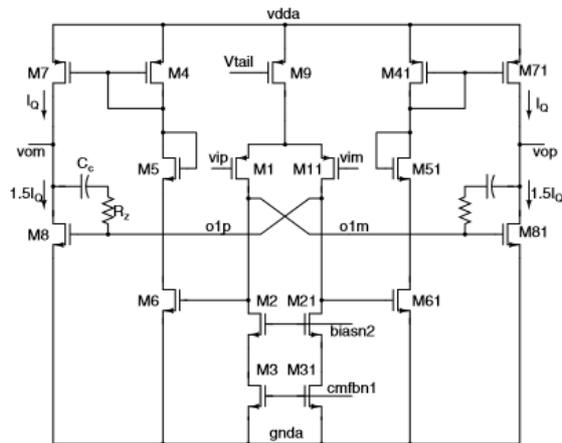


Figura 2.9 - Amplificador utilizado no primeiro integrador. Fonte: S. Pavan, N. Krishnapura, R. Pandarinathan, and P. Sankar, "A 90/ $\mu$ W 15-bit  $\Delta\Sigma$  ADC for digital audio," *ESSCIRC 2007 - Proc. 33rd Eur. Solid-State Circuits Conf.*, pp. 198–201, 2007, doi: 10.1109/ESSCIRC.2007.4430279.

O atraso excessivo é introduzido no modulador devido aos polos dos *ampops* e estes causam instabilidade no modulador, de modo a compensar este atraso excessivo é proposta a aplicação de um condensador  $C_x$ , como mostrado na Figura 2.10.

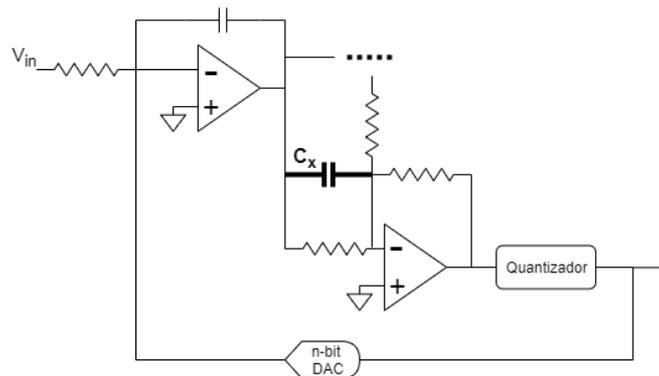


Figura 2.10 - Compensação do excesso de atraso, através do condensador  $C_x$ . Adaptada de: S. Pavan, N. Krishnapura, R. Pandarinathan, and P. Sankar, "A 90/ $\mu$ W 15-bit  $\Delta\Sigma$  ADC for digital audio," *ESSCIRC 2007 - Proc. 33rd Eur. Solid-State Circuits Conf.*, pp. 198–201, 2007, doi: 10.1109/ESSCIRC.2007.4430279.

O *Flash* ADC presente nesta arquitetura tem o comportamento descrito anteriormente. O circuito do DAC utilizado no modulador está presente na Figura 2.11, sendo composto por transístores e resistências. São necessárias resistências com um valor reduzido de modo a diminuir o atraso excessivo causado pelas capacidades parasitas.

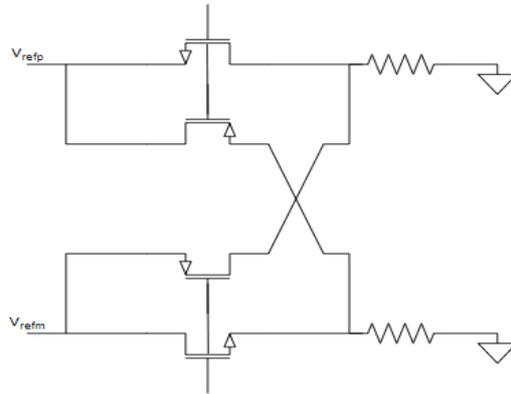


Figura 2.11 - Circuito do DAC utilizado. S. Pavan, N. Krishnapura, R. Pandarinathan, and P. Sankar, "A 90/ $\mu$ W 15-bit  $\Delta\Sigma$  ADC for digital audio," *ESSCIRC 2007 - Proc. 33rd Eur. Solid-State Circuits Conf.*, pp. 198–201, 2007, doi: 10.1109/ESSCIRC.2007.4430279.

Sigma-Delta tempo contínuo: Num delta-sigma ADC de tempo contínuo[9], Figura 2.12b, os integradores no *loop* são de tempo contínuo e não são constituídos por *switched-capacitors*, ao contrário do Sigma-Delta ADC de tempo discreto.

Os principais componentes nestas duas arquiteturas são o AAF, o modulador Sigma-Delta composto pelo S&H, um filtro de *loop*, constituído por integradores, um ADC, um DAC e um filtro decimador.

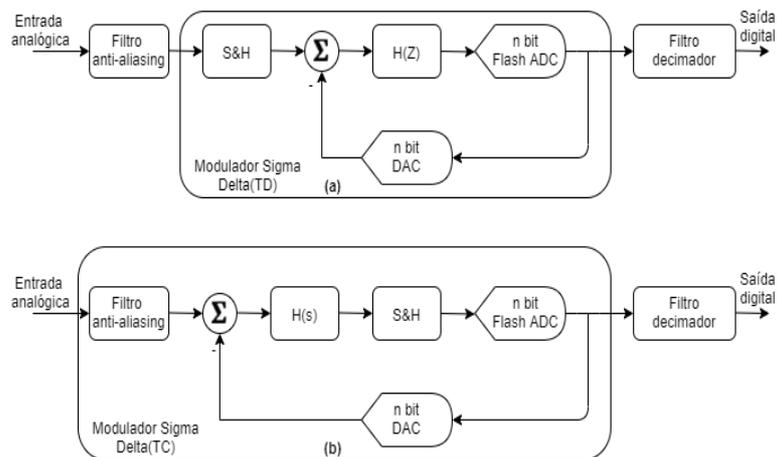


Figura 2.12 - Diagrama de blocos (a) Sigma-Delta em tempo discreto, (b) Sigma-Delta em tempo contínuo.

Fonte: J. M. De La Rosa, "Sigma-delta modulators: Tutorial overview, design guide, and state-of-the-art survey," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 58, no. 1, pp. 1–21, 2011, doi:

10.1109/TCSI.2010.2097652.

## 2.5. ADC Sigma-Delta Incremental

Relativamente ao Sigma-Delta incremental, a sua arquitetura [11] é em tudo semelhante à apresentada anteriormente para um Sigma-Delta de tempo discreto ou tempo contínuo. Consequentemente, a única diferença está relacionada com a reinicialização dos filtros.

A arquitetura geral[12] de um ADC Sigma-Delta incremental encontra-se apresentada na Figura 2.13. Esta é composta por um integrador, um quantizador que é constituído por um comparador, um DAC e um filtro digital.

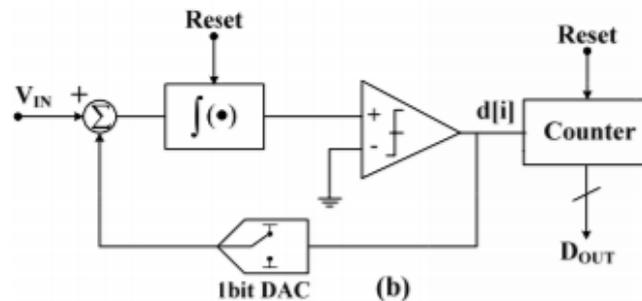


Figura 2.13 - Diagrama de blocos de um Sigma-Delta Incremental. Fonte: M. Yue, D. Wu, and Z. Wang, "Data compression for image sensor arrays using a 15-bit two-step sigma-delta ADC," *IEEE Sens. J.*, vol. 14, no. 9, pp. 2989–2998, 2014, doi: 10.1109/JSEN.2014.2321166.

Neste tipo de Sigma-Delta o filtro de *loop* e o filtro digital são reinicializados todos os ciclos [13], sendo que o modulador não executa continuamente, ou seja, é executado a uma taxa de sobre amostragem  $F_s$ , para  $M$  ciclos e ao fim de  $M$  ciclos o filtro de *loop* é reinicializado. Estes ADC têm um desempenho superior aos demais, sendo que adicionalmente não necessitam praticamente de componentes externos.

## 2.6. Arquiteturas ADC Sigma-Delta

De seguida serão abordadas algumas arquiteturas com os diferentes tipos de Sigma-Delta e com diferentes aplicações do filtro decimador de modo a possibilitar a comparação entre os seus desempenhos.

Uma possível arquitetura para um sistema sem reinicialização baseia-se no diagrama de blocos representado na Figura 2.14. Sendo esta arquitetura [8] composta por um

*multiplexer*, um circuito S&H, um modulador Sigma-Delta, um filtro decimador que é composto por um equalizador *inverse-Sinc* e um filtro de *Nyquist*.

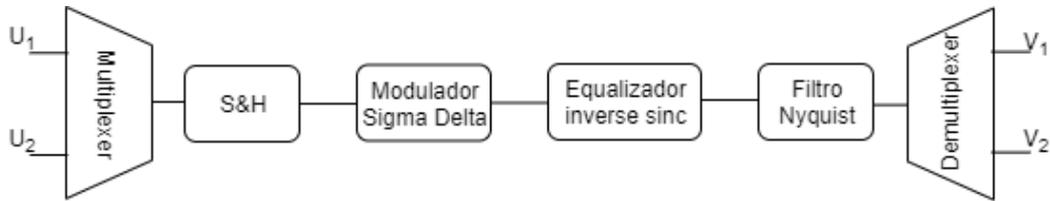


Figura 2.14 – Diagrama de blocos da arquitetura proposta. Adaptada de: R. S. A. Kumar, D. Behera, and N. Krishnapura, "Reset-Free Memoryless Delta–Sigma Analog-to-Digital Conversion," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 65, no. 11, pp. 3651–3661, 2018, doi: 10.1109/TCSI.2018.2854707.

Esta arquitetura utiliza 2 canais, sendo que os sinais são inicialmente multiplexados no *multiplexer* e são amostrados no dispositivo S&H, a uma taxa de  $\frac{F_S}{64}$ . No S&H é utilizado um método *flip around* [14], neste método inicialmente as entradas dos condensadores estão viradas para a entrada do circuito durante a fase de amostragem, posteriormente, na fase em que o circuito está em "espera" a entrada dos condensadores vira para a saída. A principal vantagem do método utilizado é a baixa potência consumida. É ainda de referir, que este circuito S&H utiliza compensação de *Miller* de modo a suportar a variação da entrada. O ruído médio quadrático introduzido no dispositivo pelo S&H é dado pela expressão (2.10).

$$N_{S\&H} = \frac{2KT}{C} + \frac{2\gamma KT}{MC_C} \quad (2.10)$$

Onde K é a constante de Boltzman, T representa a temperatura no dispositivo, em Kelvin, C é o condensador presente no S&H utilizado método *flip around*,  $\gamma$  é o fator de ruído do transistor e  $C_C$  é condensador utilizado para a compensação de *Miller*.

O modulador Sigma-Delta é incremental e de tempo discreto, estando o seu diagrama de blocos na Figura 2.15.

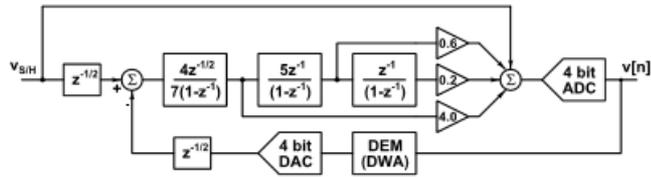


Figura 2.15 - Diagrama de blocos do modulador Sigma-Delta de 3ª ordem. Fonte: R. S. A. Kumar, D. Behera, and N. Krishnapura, "Reset-Free Memoryless Delta-Sigma Analog-to-Digital Conversion," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 65, no. 11, pp. 3651–3661, 2018, doi: 10.1109/TCSI.2018.2854707.

Este modulador possui um *feedforward path* no filtro de *loop* de modo a diminuir a presença de componentes do sinal de entrada à saída, resultando em Função Transferência do sinal (STF)=1. O *feedforward path* é importante no aumento da linearidade dos *ampops* presentes no circuito. Ademais é necessário, pois o filtro decimador utilizado e explicado de seguida força a um STF=1. O ruído devido ao modulador é dado pela expressão (2.11).

$$N_{MSD} = \frac{2KT}{MC_{in}} \quad (2.11)$$

Onde  $C_{in}$  representa o condensador introduzido à entrada do modulador.

O filtro decimador é constituído por dois componentes, o equalizador *inverse Sinc* e o filtro de *Nyquist* a uma taxa  $F_s$ . O equalizador *inverse Sinc* é utilizado no filtro decimador para remover a memória introduzida no sistema pelo S&H utilizado. O inverso da função transferência do S&H apenas é realizado para a banda do sinal entre 0 e um pouco além de  $\frac{F_s}{2M}$  [15], pois o inverso da função transferência é instável e não linear. Para o resto do sinal não é necessário fazê-lo pois o filtro de *Nyquist* atenua o sinal fora da banda. O filtro de *Nyquist* utilizado é um filtro cosseno, que quando amostrado a uma taxa de  $\frac{F_s}{M}$  a sua resposta é obtida sem memória.

A arquitetura [16] apresentada na Figura 2.16 representa um modulador Sigma-Delta ADC incremental de tempo contínuo de 3ª ordem com 3 bits, composto por 3 integradores, um quantizador constituído por um DAC e um ADC, e ainda um DAC de entrada.



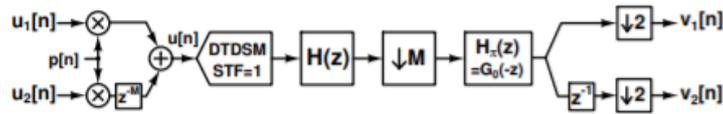


Figura 2.17 - Modulador Sigma-Delta de 2 canais com filtro  $\pi$ -shifted. Fonte: R. S. A. Kumar and N. Krishnapura, "A 2-Channel ADC Using a Delta-Sigma Modulator Without Reset & a Modulated-Sinc-Sum Filter," *ESSCIRC 2019 - IEEE 45th Eur. Solid State Circuits Conf.*, pp. 365–368, 2019, doi: 10.1109/ESSCIRC.2019.8902610.

O modulador Sigma-Delta tem  $STF=1$  devido à utilização de um *feedforward path*. Este é implementado com 3 integradores em cascata, sendo que o primeiro utiliza um *telescopic cascode opamp* com um *switched capacitor*, os restantes integradores utilizam uma versão simplificada do mesmo.

O filtro decimador ( $H(z)$ ) é composto por 2 filtros *raised-cosine* seguidos de 2 sub-amostragens. O primeiro filtro *raised-cosine* é de ordem 256 e é subamostrado por 2 e o segundo é de ordem 16 e subamostrado por 2. Como é possível observar na Figura 2.17, é utilizado um filtro  $\pi$ -shifted ( $H_\pi = G_0(-z)$ ), sendo que  $G_0$  é o equivalente ao circuito S&H, precedido do modulador e por fim o filtro decimador.

Esta arquitetura foi implementada com uma tecnologia CMOS 180nm que resultando num protótipo com uma área de 3,78mm<sup>2</sup> e dissipando 1,53mW. Com a análise deste sistema obtém-se um SNR de 94,4 dB, um SNDR DE 93dB e um DR de 98,5.

A arquitetura [19] sugerida consiste num ADC Sigma-Delta incremental de tempo discreto que opera para  $N = OSR$  ciclos de relógio a uma frequência  $F_s$ . A Figura 2.18 representa o modulador Sigma-Delta implementado com um filtro Col de 3<sup>a</sup> ordem.

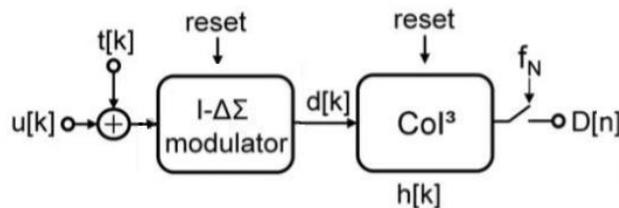


Figura 2.18 - Diagrama de blocos do modulador com o filtro decimador Col. Fonte: P. Vogelmann, M. Haas, and M. Ortmanns, "A 1.1mW 200kS/s incremental  $\Delta\Sigma$  ADC with a DR of 91.5dB using integrator slicing for dynamic power reduction," *Dig. Tech. Pap. - IEEE Int. Solid-State Circuits Conf.*, vol. 61, pp. 236–238, 2018, doi: 10.1109/ISSCC.2018.8310271.

Os maiores problemas de linearidade ou de ruído são provenientes do primeiro integrador do filtro de *loop*, portanto é sugerida uma reconfiguração dinâmica que divide o

integrador em 4 partes, independentes umas das outras, sendo assim possível ativar e desativar partes do integrador durante as conversões. Como exemplo, está representado na Figura 2.19 uma das partes do primeiro integrador.

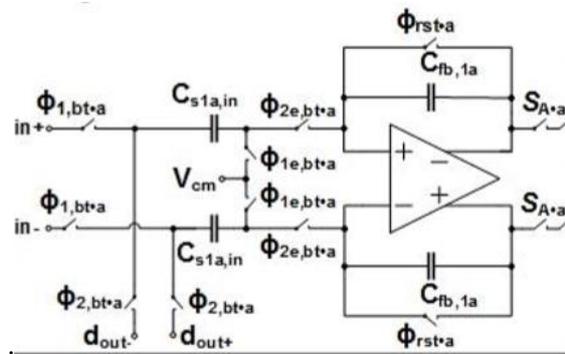


Figura 2.19 - Circuito de uma das 4 partes do primeiro integrador. Fonte: P. Vogelmann, M. Haas, and M. Ortmanns, "A 1.1mW 200ks/s incremental  $\Delta\Sigma$  ADC with a DR of 91.5dB using integrator slicing for dynamic power reduction," *Dig. Tech. Pap. - IEEE Int. Solid-State Circuits Conf.*, vol. 61, pp. 236–238, 2018, doi: 10.1109/ISSCC.2018.8310271.

O filtro Col é o filtro digital que vai gerar sinal final que funciona a  $F_{N=} = \frac{F_s}{OSR}$  que é composto por um conjunto de integradores ligados. Este filtro apenas vai contribuir para a função de transferência da conversão. Após a digitalização o modulador e o filtro são reinicializados.

Esta arquitetura foi implementada numa tecnologia 180nm CMOS, sendo a sua área 0,363mm<sup>2</sup>. A aplicação desta técnica de separação leva a uma redução da potência de 1,65mW para 1,098mW, chegando assim a um SNR de 88,2 dB, um SNDR de 86,5dB e um DR de 91,5 dB. O  $FoM_s$  obtido é de 166,2 dB.

Aplicações de instrumentação e medição necessitam de alta resolução e linearidade, como tal, os conversores Sigma-Delta são uma boa solução. Na Figura 2.20 está representado o diagrama de blocos de uma arquitetura [20] que tem como objetivo principal a redução da potência consumida.

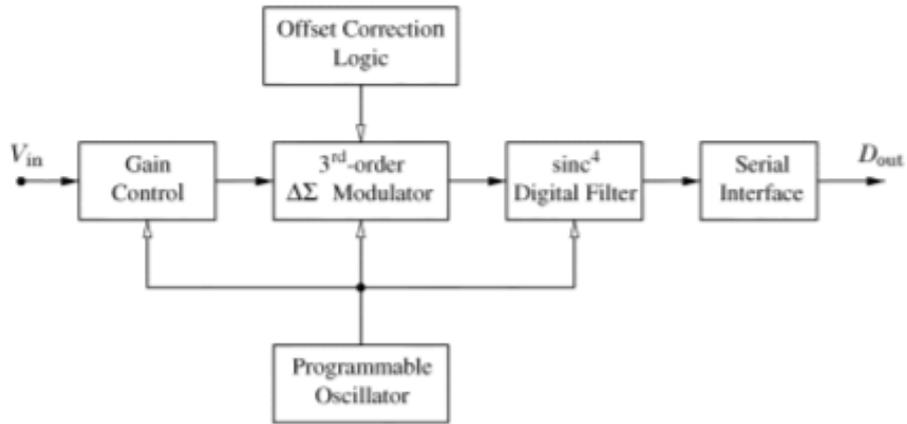


Figura 2.20 - Diagrama de blocos do conversor incremental proposto. Fonte: V. Quiquempoix *et al.*, "A low-power 22-bit incremental ADC," *IEEE J. Solid-State Circuits*, vol. 41, no. 7, pp. 1562–1571, 2006, doi: 10.1109/JSSC.2006.873891.

O bloco de ganho de controlo torna-se importante pois o sinal de entrada tem de ser atenuado para não sobre carregar o filtro de *loop*. Para reduzir o ganho e simultaneamente aumentar a linearidade é utilizado o circuito da Figura 2.21.

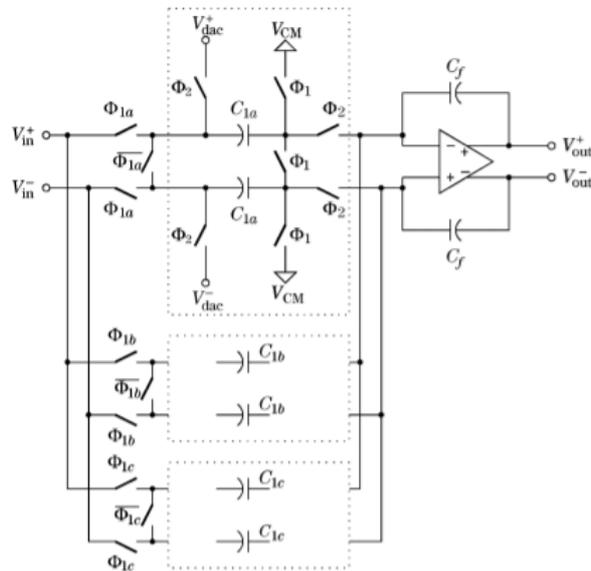


Figura 2.21 - Arquitetura do bloco de controlo. Fonte: V. Quiquempoix *et al.*, "A low-power 22-bit incremental ADC," *IEEE J. Solid-State Circuits*, vol. 41, no. 7, pp. 1562–1571, 2006, doi: 10.1109/JSSC.2006.873891.

A lógica de correção de *offset* é a versão geral da estabilização de *Chopper*, baseado num algoritmo fractal [21] que cancela em parte o *offset* do sinal de entrada após estar concluída a sua amostragem.

O modulador Sigma-Delta utilizado é de 3ª ordem com uma configuração de baixa distorção e de modo a ser necessária menor linearidade da parte dos *ampops* são usados integradores *switched-capacitor*. Nesta arquitetura o  $OSR = 512$ .

O filtro digital é um filtro *Sinc* de quarta ordem constituído por diferenciadores, integradores e elementos de escala. Por outro lado, nesta implementação, após um sinal ser integrado é utilizado um contador programável, que produz um sinal diferencial e amostra-o a uma frequência mais baixa[22].

Esta arquitetura [23], constituída por um ADC Sigma-Delta incremental, de tempo contínuo de dois andares., tem o seu diagrama de blocos representado na Figura 2.22.

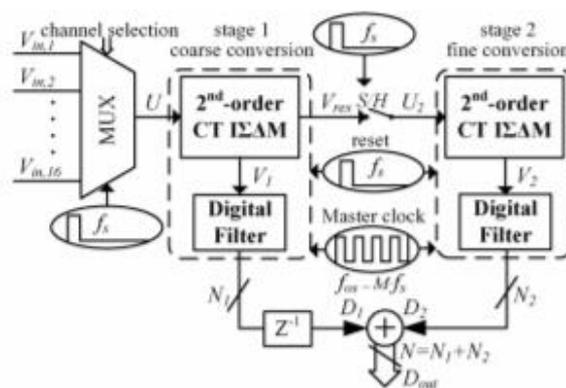


Figura 2.22 - Diagrama de blocos do modulador Sigma-Delta de 2 andares com filtro digital. Fonte: h and S. Pavan, "A 14 Bit Dual Channel Incremental Continuous-Time Delta Sigma Modulator for Multiplexed Data Acquisition," *Proc. IEEE Int. Conf. VLSI Des.*, vol. 2016-March, pp. 230–235, 2016, doi: 10.1109/VLSID.2016.21.

O modulador Sigma-Delta utilizado é de segunda ordem e é constituído por uma configuração Col com *feedforward* (CIFF) com uma topologia *Input-Feedforward* (IFF) e utiliza um quantizador de 1 bit. O filtro digital, nesta arquitetura, é bastante semelhante ao filtro de *loop* do modulador, sendo que também se baseia num filtro Col. Podemos ver esta configuração na Figura 2.23.

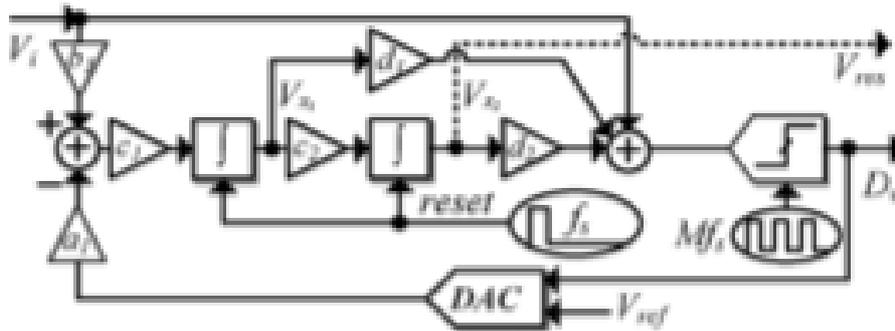


Figura 2.23 - Modulador Sigma-Delta de tempo contínuo de 2 andares

Inicialmente o *multiplexer* seleciona um dos sinais de entrada e os conversores são reinicializados. De seguida o sinal é amostrado intervaladamente pelos conversores Sigma-Delta pelas 2 etapas de conversão. A primeira etapa de conversão processa o sinal proveniente do *multiplexer*, durante  $M$  ciclos, de seguida o filtro digital extrai os bits menos significativos e o modulador e o filtro são reinicializados. Deste processo resulta um resíduo analógico que entra no S&H e de seguida entra na segunda etapa de conversão. Este resíduo passa por um processo semelhante no modulador, sendo que neste caso, após os mesmos  $M$  ciclos são extraídos os bits mais significativos. Por fim o modulador e o filtro digital são reinicializados.

Este sistema tem 16 canais, cada um com 200Hz de largura de banda. De modo a obter o tempo máximo pertencente atribui-se a cada canal uma frequência de amostragem maior que a frequência de *Nyquist*. A expressão 2.12 e 2.13 representam o tempo máximo dedicado a cada canal e a frequência efetiva de conversão.

$$T_s = \frac{1}{200 * 2,5 * 16} = 125\mu s \quad (2.12)$$

$$f_s = 8kHz \quad (2.13)$$

Para o resíduo analógico proveniente da primeira etapa não ser maior do que a entrada do segundo modulador suporta, a saída do último integrador do modulador, que amostra o resíduo, assume que a entrada é constante.

No filtro de *loop* é constituído por um *feedforward coefficient summation* que realiza as somas ponderada e um quantizador de 1 bit, como representado na Figura 2.24.

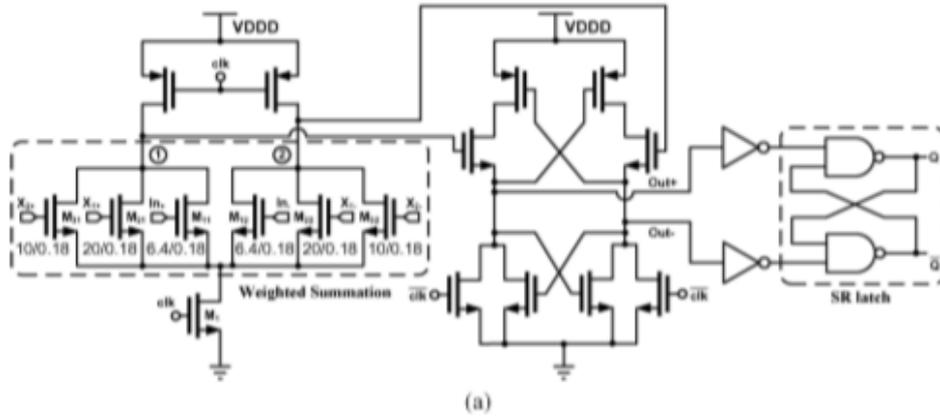


Figura 2.24 - Arquitetura do Feedforward coefficient summation com quantizador de 1 bit

O DAC do modulador é controlado por 2 grupos de interruptores e duas resistências de *feedback* e as suas tensões de referência (positiva e negativa) são comutadas pelas 2 saídas do quantizador.

O filtro digital incorporado no conversor é a soma da Col que processa as M amostras provenientes do modulador sendo o resultado à saída as somas ponderadas do sinal à entrada do filtro. A expressão (2.14) representa a função transferência do filtro digital.

$$f_t = \left( \frac{z^{-2}}{(1 - z^{-1})^2} + \frac{z^{-1}}{2(1 - z^{-1})} \right) \frac{2}{M(M - 1)} \quad (2.14)$$

A arquitetura implementada e testada tem uma tecnologia 180nm CMOS com, e a sua área é 0,337mm<sup>2</sup>. A potência consumida é 34,8μW e os resultados obtidos para o SNR, SNDR e DR são, respetivamente, 76,6dB, 75,9dB e 85,5dB.

Na arquitetura [24] baseia-se num modulador Sigma-Delta incremental de tempo contínuo de 3ª ordem com 15 bits de resolução.

Este conversor é constituído pelo modulador Sigma-Delta que possui um filtro de *loop* constituído por integradores, 2 DACs, um de *feedback* e um de assistência que se comporta como um condutor de corrente e um comparador de 1 bit. Para além do modulador, o filtro digital é um filtro Col decimador de 3ª ordem com 3 integradores. O modulador foi implementado com uma configuração ClIFF e os seus integradores são implementados com um *feedforward compensated* OTA.

A arquitetura [24] consiste num modulador Sigma-Delta incremental de tempo contínuo. A arquitetura é composta por 2 DACs um comporta-se como condutor de corrente e o outro é um DAC de *feedback* resistivo. Tem ainda, um transcondutor que controla a

corrente na saída do primeiro OTA e um comparador de 1 bit. Para além do modulador, o filtro digital é um filtro Col decimador de 3ª ordem com 3 integradores. Esta arquitetura foi implementada com um modulador Sigma-Delta de incremental de tempo contínuo de 3ª ordem com configuração CIFF que opera com uma frequência de amostragem de 6,144MHz.

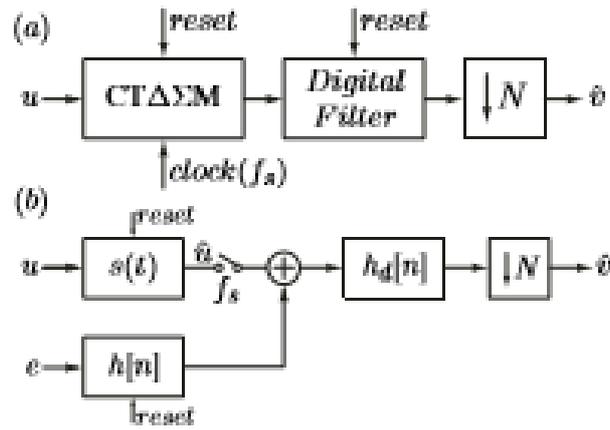


Figura 2.25 - a) Diagrama de blocos da arquitetura; b) Versão modelada da mesma arquitetura

Na Figura 2.25a está representado o digrama de blocos do modulador e do filtro. Na Figura 2.25b está representado uma versão modelada em que  $s(t)$  é a resposta em tempo contínuo correspondente ao STF e o  $e$  representa o ruído proveniente do quantizador. O filtro de *loop* é realizado com uma configuração CIFF e a sua função transferência é do tipo da expressão (2.15) e é reinicializado após  $N$  ciclos.

$$tf_l = \frac{k_1}{s} + \frac{k_2}{s^2} + \frac{k_3}{s^3} \quad (2.15)$$

O filtro digital é um filtro decimador Col constituído por 3 integradores de função transferência  $\frac{z}{z-1}$ .

A Função Transferência do Ruído (NTF) do modulador, expressão (2.16), e a função transferência do filtro Col, expressão (2.17), podem ser combinados num único filtro com a função transferência da expressão (2.18).

$$NTF = \frac{(z-1)^3}{D(z)} \quad (2.16)$$

$$tf_{col} = \frac{z^3}{(z-1)^3} \quad (2.17)$$

$$tf = \frac{z^3}{D(z)} \quad (2.18)$$

Como se pode perceber pela combinação das funções transferência os zeros do NTF são cancelados pelos polos do filtro decimador.

A Figura 2.26 representa a implementação do primeiro integrador que é implementado com um *feedforward compensated* OTA. A única diferença deste integrador para os outros dois é o facto de o segundo e o terceiro terem à entrada um par de transístores NMOS.

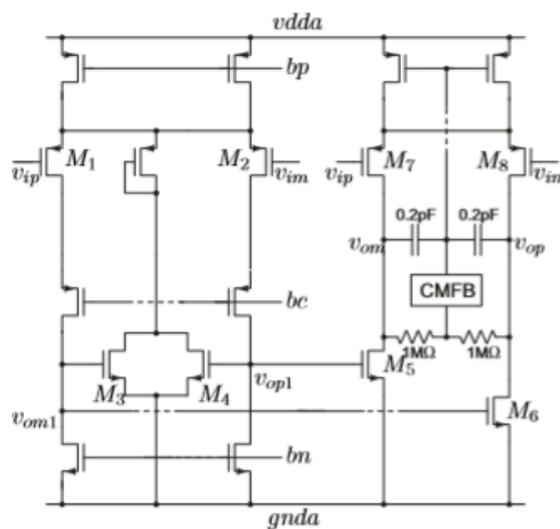


Figura 2.26 - Arquitetura Ota utilizada no primeiro integrador

Esta arquitetura foi implementada com uma tecnologia 180nm CMOS, o ADC funciona a uma frequência de amostragem de 12kS/s e tem uma área de 0,35mm<sup>2</sup>. O valor obtido de SNR e SNDR é 85,1 para ambos e a potência dissipada é 55μW.

O conversor proposto nesta arquitetura [25] é baseado num modulador incremental Sigma-Delta de segunda ordem. Este é um conversor com 15 bits de resolução e funciona a 45,2MHz.

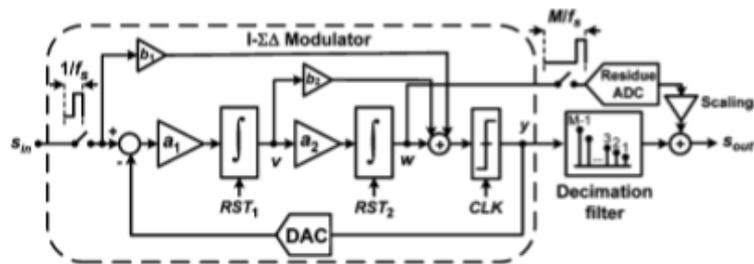


Figura 2.27 - Diagrama de blocos do modulador de 2ª ordem com o filtro decimador. Fonte: A. Agah, K. Vleugels, P. B. Griffin, M. Ronaghi, J. D. Plummer, and B. A. Wooley, "A high-resolution low-power incremental  $\Sigma\Delta$  ADC with extended range for biosensor arrays," *IEEE J. Solid-State Circuits*, vol. 45, no. 6, pp. 1099–1110, 2010, doi: 10.1109/JSSC.2010.2048493.

O modulador apresentado na Figura 2.27 é um incremental Sigma-Delta com *extended range*, ou seja, é feita uma estimativa do erro residual que depois é combinada com a saída do filtro decimador assim é possível cancelar o erro de quantização resultante do processo do filtro decimador.

Para a implementação do modulador, foi utilizado uma tecnologia 180nm CMOS e tem uma área de 3,5mm<sup>2</sup>. A frequência de amostragem a que o ADC opera é 45,2MHz. Ao ser testada a arquitetura os valores de DR, SNR, SNDR e SFDR são, 90,1, 89,1, 86,3 e 84,5, respectivamente.

Na arquitetura [26] é sugerida um Sigma-Delta Incremental de tempo discreto com um sinal de entrada contínuo que é amostrado pelo *switched-capacitor* do primeiro integrador do modulador. Este modulador opera a uma taxa de amostragem  $F_s$  que será  $M$  vezes superior à saída digital e decimada que é obtida a uma taxa de *Nyquist*,  $F_N$ . É produzido um sinal digital à saída em cada ciclo, que necessita ser decodificado através de um filtro digital.

De modo a conseguir reduzir o SQNR, é aplicada uma fase de recuperação (RP) que consiste em descartar o início de cada fase de conversão, para isso a entrada do modulador é desligada do sinal de entrada e ligada ao limite do quantizador. De modo a perceber o número de ciclos necessários para se obter os resultados pretendidos foi utilizada uma arquitetura de 3ª ordem que necessita de um  $OSR = 160$  para obter um  $SNR \geq 94$ . Foram feitas simulações para perceber como o erro se comportava durante um único ciclo de conversão, para isso foram definidos 3 valores distintos de RP, cada um começando R ciclos antes do fim da conversão. As simulações mostraram que sem RP o erro de quantização diminui continuamente, mas para qualquer valor de R houve uma descaída no início de RP, mas de seguida houve uma diminuição constante até ao final de

uma conversão. Ao escolher a duração de RP é necessário ter também em atenção a atenuação da potência do sinal. Um alto valor de R levaria a uma grande influência na amplitude do sinal, pois durante o RP apenas é atenuado o ruído de quantização e não o ruído do circuito.

A introdução da RP leva à possibilidade de o circuito ter um MSA mais elevado o que permite ter uma potência dissipada inferior. Isto deve-se ao facto de o sinal de entrada estar desligado e assim não há nenhuma potência de sinal adicional injetada no sistema, consequentemente, durante os ciclos antes da Fase de Recuperação a amplitude poderá ser bastante maior sem originar a sobrecarga. Através das simulações é concluído que com a presença de RP a amplitude de entrada para o pico de SNQR é estendida 2dB. A introdução da RP também é uma vantagem para a quantização multibit, visto que durante esta fase não está nenhum sinal presente na entrada o que leva a que apenas exista um pequeno erro associado à saída do último integrador

Uma desvantagem do uso de RP é que ao aumentar o MSA do circuito isso leva a um possível aumento da oscilação na saída do primeiro integrador. Essa oscilação pode ter consequências como a saturação do integrador, o que pode afetar significativamente o desempenho. Para reduzir os efeitos do aumento do MSA podem ser definidos novos valores para os coeficientes do circuito de modo que a oscilação seja reduzida. Aumentando o condensador de integração pode ser obtida esta nova escala.

Tendo sido esta arquitetura implementada com uma tecnologia 180nm CMOS com um modulador Sigma-Delta incremental de 3ª ordem, houve um aumento de 1.9dB no SNR passando 95.6dB, sem RP, para 97,5dB, com RP. Para além disso também foi possível um aumento da tensão de entrada, de 2V para 2,5V.

## 2.7. Arquiteturas ADC Sigma-Delta com decodificação Ótima

Uma outra solução de decodificação, é apresentada no artigo [27] onde inicialmente é utilizado um Sigma-Delta incremental de primeira ordem de modo a explicar o método. É assumido um  $V_{in} \in [-1,1]$ , que permanece constante durante 1 conversão, e a saída do quantizador,  $q \in \{-1,1\}$ . Entre cada conversão o integrador é reinicializado, através de  $u_1 = 0$ ,  $u_1$  a saída do integrador. A saída do integrador,  $u_1$ , pode ser obtida através da expressão simplificada (2.19), onde  $S_n$  é a soma de  $q[0]$  a  $q[n - 1]$ .

$$u_1(n) = n - V_{in} * S_n, n \geq 1 \quad (2.19)$$

Por sua vez a saída digital  $q[n]$  é definida por (2.20)

$$\begin{cases} 1, V_{in} \geq \frac{S_n}{n} \\ -1, V_{in} < \frac{S_n}{n}, n \geq 1 \end{cases} \quad (2.20)$$

De modo a codificar uma entrada  $V_{in}$  pode ser adicionada uma rampa  $n * V_{in}$ , rampa de entrada, que comparada com a inclinação formada pela ligação dos pontos  $(0,0)$  com  $(n, S_n)$ , rampa correspondente a  $\frac{S_n}{n}$ , tem como resultado a saída do modulador. Sendo assim, caso o valor da rampa de entrada seja maior então o  $q[n] = 1$ , por outro lado caso o valor seja inferior, nesse caso  $q[n] = -1$ .

De acordo com este método se  $q[n] = 1$  então  $V_{in} \geq \frac{S_n}{n}$  isto significa que neste momento o limite inferior será este valor de  $\frac{S_n}{n}$ . Por outro lado, se  $q[n] = -1$  então  $V_{in} < \frac{S_n}{n}$ , assim o limite superior será este valor  $\frac{S_n}{n}$ . Aplicando este método à saída do modulador é possível ir reduzindo o intervalo de  $V_{in}$  descobrindo assim o limite máximo,  $lb_{max}$ , e o limite mínimo,  $lb_{min}$ .

Este método permite atingir uma potência média do ruído inferior e melhorar a precisão do sistema. Comparando o erro médio quadrático (MSE) deste algoritmo de decodificação com o de um filtro Col ou Sinc de segunda ordem, este é o que possui o menor resultado de MSE. Uma desvantagem é que a saída do primeiro integrador é afetada pelo ruído térmico do circuito.

A decodificação de um modulador de primeira e de segunda ordem, é feita através dos valores calculados de  $lb_{max}$  e  $lb_{min}$ . Após os cálculos podem existir dois casos distintos, o primeiro em que  $lb_{max}$  seria menor que  $lb_{min}$ , e neste caso é definida uma região entre  $lb_{max}$  e  $lb_{min}$  que é uma estimativa de onde se encontra o sinal de entrada. Por outro lado, na situação em que  $lb_{max}$  é maior que  $lb_{min}$  o valor de  $V_{in}$  pode estar longe dos limites, por isso, para a decodificação é usado o limite que passou por um maior número de ciclos de quantização. Assim sendo esta decodificação permite manter um ruído de quantização baixo e uma boa média de ruído térmico.

Para a análise do decodificador proposto foi utilizado um modulador de segunda ordem com tecnologia 180nm e que ocupa uma área de  $0.15\text{mm}^2$ . Utilizando OSR de 150 o ruído térmico é de  $31\mu V_{rms}$ , para o decodificador proposto, o que significa uma redução em comparação com um filtro Col<sup>2</sup> que obteve um ruído térmico de  $63\mu V_{rms}$  ou com

um  $\text{Sinc}^3$  com o qual o ruído térmico é de  $102,1\mu V_{rms}$ . O SNR também mostra uma melhoria relativamente aos filtros  $\text{Col}^2$  e  $\text{Sinc}^3$  de 6,1dB e 10,3dB.

Na arquitetura [28] é feita a correspondência entre a terminologia usada nas arquiteturas de *design* de circuitos e em [29], como se pode ver na Figura 2.28 que descreve a teoria da quantização. Nesta teoria o modulador é descrito como um codificador que produz uma sequência binária com tamanho correspondente ao intervalo de quantização do sinal de entrada. O filtro que corresponde ao decodificador produz uma estimativa do sinal de entrada, através do ponto médio do intervalo de quantização que o filtro possui na sua saída.

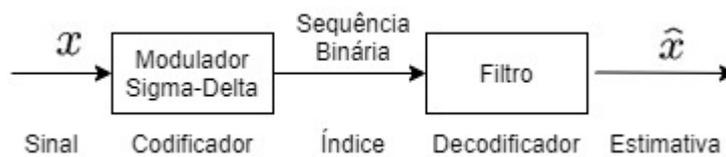


Figura 2.28 - Correspondência entre a terminologia utilizada em *Design* de circuitos e a implementada em [28]. Adaptada de: S. Kavusi, H. Kakavand, and A. El Gamal, "On Incremental Sigma – Delta Modulation," vol. 53, no. 5, pp. 1004–1015, 2006.

Inicialmente a arquitetura é estudada para um modulador Sigma-Delta de primeira ordem com sinal de entrada  $x \in [0,1]$  e o valor limite do comparador é igual a 1. O sinal de entrada é integrado no modulador criando uma rampa com inclinação proporcional à entrada. O valor desta rampa em cada instante vai ser comparada com o valor 1 e o valor à saída do comparador segue o seguinte critério: Caso a amostra da rampa seja superior a 1 a saída terá o valor 1, e este será subtraído ao valor da rampa, caso contrário a saída terá o valor 0. Assim o modulador faz a previsão do valor da rampa contando o número de uns e subtraindo-os da sua previsão através do *feedback loop*.

A sequência de comparações feitas entre o sinal à entrada do comparador e 1, no modulador Sigma-Delta é equivalente a um codificador que produza bits correspondentes para as mesmas comparações. Estas comparações permitem definir um limite máximo,  $UB_{max}$  e um limite mínimo,  $LB_{min}$ .

Para encontrar os pontos de transição entre os intervalos de quantização é definido um codificador equivalente que produz a mesma sequência de comparações que o comparador faz entre a saída do integrador e 1, mas através da previsão mostrada na expressão (2.21) para a rampa equivalente  $nx$ , em que  $n$  corresponde ao instante de tempo.

$$1 + \sum_{i=0}^{n-1} b(i) \quad (2.21)$$

O algoritmo sugerido tem como objetivo seguir os critérios de um filtro ótimo, o que significa que dada uma sequência produzida pelo modulador o filtro vai gerar o ponto médio do intervalo de quantização. Para confirmar que o algoritmo vai de encontro a um filtro ótimo é comprovado que os limites máximo e mínimo são pontos de transição. Sendo  $m$ , os instantes de amostragem, até  $m - 1$  os limites máximo,  $UB$  e mínimo  $LB$ , são pontos de transição do intervalo  $x$ . Para os restantes instantes sendo que apenas poderá haver um novo ponto entre  $UB$  e  $LB$ , e assumindo  $p$  como valor previsto é seguida a seguinte lógica: se  $\frac{p}{m} \notin (LB, UB)$  então o intervalo de quantização mantém-se  $(LB, UB)$  e  $UB$  e  $LB$  são pontos de transição. Por outro lado, se  $\frac{p}{m} \in (LB, UB)$  o valor de  $UB$  e  $LB$  será alterado consoante a sequência do modulador no instante  $m$ , se for 1 é atualizado o valor de  $LB$  para  $\frac{p}{m}$  e se for 0 é atualizado o valor de  $UB$  para  $\frac{p}{m}$ , sendo os novos intervalos de quantização  $(\frac{p}{m}, UB)$  e  $(LB, \frac{p}{m})$ , respetivamente.

Da análise da MSE do filtro ótimo, comparando com outros filtros como o triangular é obtido um valor inferior para a MSE do filtro ótimo estudado. O erro máximo atingido pelo filtro ótimo também é bastante inferior que os restantes filtros lineares, para a mesma taxa de amostragem, o que permite atingir o mesmo erro absoluto com taxas de amostragem inferiores.

Para o estudo do modulador Sigma-Delta de segunda ordem é utilizado um modulador que consiste em 2 integradores e 1 comparador as suas saídas são realimentadas na entrada dos integradores. Ao contrário do modulador de primeira ordem, neste caso o comparador tem o valor de 0 e o sinal de entrada  $x \in [-1,1]$ .

Para encontrar os pontos de transição de modo a determinar os intervalos de quantização induzidos pelo modulador é novamente utilizado um codificador. A comparação que é feita entre a saída do segundo integrador e 0 tem a mesma sequência que a previsão mostrada na expressão (2.22). Ambas as sequencias representam as comparações presentes em (2.23) em que  $b$  representa a sequência à saída do comparador.

$$\sum_{i=1}^{n-1} (n - i + 1) * b(i) \quad (2.22)$$

$$x \leq 2 \left( \sum_{i=1}^{n-1} \left( \frac{n-i+1 * b(i)}{n^2+n} \right) \right) \quad (2.23)$$

No caso do modulador de segunda ordem, ao contrário do modulador de primeira ordem nem todos os valores inteiros que pertencem ao intervalo de quantização são pontos de transição, por isso neste caso é utilizado o *Directed acyclic graph* (DAG) para obter os pontos de transição.

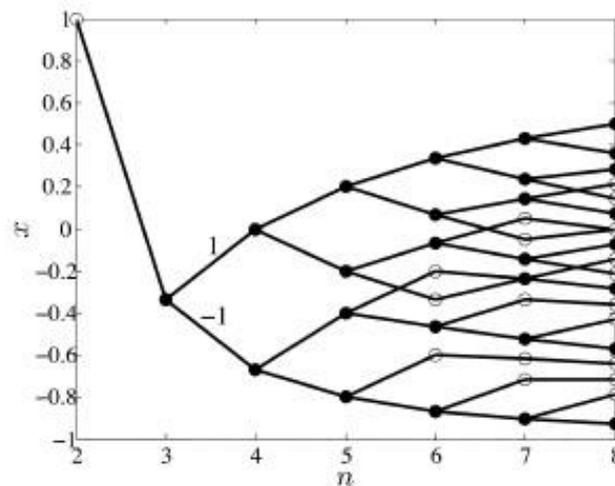


Figura 2.29 - Directed acyclic graph para o modulador Sigma-Delta de segunda ordem. Fonte: S. Kavusi, H. Kakavand, and A. El Gamal, "On Incremental Sigma – Delta Modulation," vol. 53, no. 5, pp. 1004–1015, 2006.

Este gráfico permite limitar os valores de  $x$  bem como identificar os pontos de transição (os pontos sólidos marcados, no gráfico da Figura 2.29).

Em relação ao algoritmo sugerido para o modulador de primeira ordem, os resultados ao nível da MSE, para o modulador de segunda ordem com o filtro ótimo, são ainda melhores, quando comparados com os restantes filtros lineares.

Este algoritmo tem vantagens em diferentes tipos de aplicações, em sensores de imagem ou de temperatura, onde é necessário baixo consumo, neste caso, com o Sigma-Delta Incremental é possível obter cerca de menos  $100\mu W$  de consumo no modulador quando comparado com o modelo convencional e adicionando o filtro ótimo, este consome apenas  $10nW$ . Normalmente os filtros ótimos têm maior potência dissipada do que os filtros

lineares devido ao número de cálculos por amostra, o que neste caso não se confirma, devido ao filtro ótimo operar a um quinto da frequência. Também tem vantagens para sinais de banda limitada onde a maior diferença entre o modelo convencional e o incremental Sigma-Delta é as taxas a que operam cada um dos modelos. Enquanto no convencional o sinal de entrada é amostrado a uma taxa  $f_1$ , bastante superior à taxa de *Nyquist* do sinal de entrada, e o modulador opera à mesma taxa  $f_1$ . O sinal de saída é amostrado a uma taxa  $f_2$  que é ligeiramente superior à taxa de *Nyquist*, originando uma taxa de sobre amostragem de  $\frac{F_1}{F_2}$ , no sistema incremental o sinal inicialmente é amostrado a uma taxa  $F_2$  e o modulador Sigma-Delta opera à taxa  $F_1$ . Nesta aplicação existe uma grande vantagem ao nível do SNDR que é 7 dB mais elevado para o Incremental Sigma-Delta.

## 2.8. Tabela de comparação de resultados

A Tabela 2.2 engloba as métricas e mais valores relevantes para análise e comparação das várias arquiteturas descritas no capítulo anterior.

Para análise, é importante ter em conta a área ocupada pelo dispositivo, pois é relevante este tipo de dispositivo ter um tamanho reduzido. Neste ponto a arquitetura [20], é a que melhor cumpre este requisito, sendo que as arquiteturas [17], [24] e [25], também têm áreas reduzidas. Outro parâmetro que hoje em dia também é preciso ter bastante atenção é a potência dissipada, sendo que neste caso claramente se destacam as arquiteturas, [10], [24] e [25]. O FOM também é importante na análise da eficiência, sendo que um  $FOM_w$  mais baixo leva a maior eficiência. Neste aspeto, a arquitetura [10], mostra-se bastante melhor que outras arquiteturas, contudo as arquiteturas [24] e [25] também têm valores relativamente reduzidos. Quanto ao  $FOM_s$ , quanto maior for maior será a eficiência, logo, as arquiteturas [19] e [20] serão, avaliando este parâmetro, mais eficientes que a arquitetura [14]. É também importante analisar métricas como o SNR e o SNDR, tendo em conta que um maior valor de SNR ou SNDR significa uma maior "quantidade" de sinal em relação ao ruído ou em relação ao ruído e distorção respetivamente. Nestas métricas a arquitetura que mais se destaca é a [19], no entanto as arquiteturas [10], [20] e [26] também apresentam valores relativamente altos.

Feita esta análise, é importante avaliar qual arquitetura que tem uma relação mais favorável entre todos estes valores, ou seja, tentar perceber qual arquitetura consegue

conjugar melhor o facto de ser necessária área reduzida, pouca dissipação de potência, eficiência e um boa relação sinal-ruído. As arquiteturas que melhor conseguem cumprir todos esses requisitos são a arquitetura [10] e [25].

Tabela 2.2 - Valores obtidos para as arquiteturas descritas no Estado da Arte

	[10]	[8]	[16]	[18]	[19]	[23]	[24]	[25]
Arquitetura (Ordem)	SDTC 3 <sup>a</sup>	ISDTD 3 <sup>a</sup>	SDTC 3 <sup>a</sup>	SDTD 3 <sup>a</sup>	ISDTD 3 <sup>a</sup>	ISDTC 2 <sup>a</sup>	ISDCT 3 <sup>a</sup>	ISD 2 <sup>a</sup>
Tecnol. (nm)	180	180	120	180	40	180	180	180
Área (mm <sup>2</sup> ) (Circuito integrado)	0,72	3,86	0,11	3,78	0,074	0,337	0,35	3,5
Freq. de amostragem (MHz)	3,072	32	104	6,144	30	-	6,144	45,2
Resolução ENOB	15	4	10	-	15	14	14	15
DR (dB)	-	-	-	-	14,1	-	-	-
SNR (dB)	93,5	84,2	62	98,5	91,5	85,5	-	90,1
SNDR (dB)	92,5	82,5	60	94,4	88,2	76,6	85,1	89,1
SFDR (dB)	90,8	80,1	58	93	86,5	75,9	85,1	86,3
Pot. Dissipada	-	-	-	-	-	-	97	94,5
FOM <sub>w</sub> (pj/conv-step)	90	32,7	3	1,53	1,098	34,8	55	38,1
FOM <sub>s</sub> (dB)	$\mu$ W	mW	mW	mW	mW	$\mu$ W	$\mu$ W	mW
	0,049	2,91	-	-	-	0,85	0,31	1,46
	-	154,3	-	170,1	171,1	-	-	-

## SIGMA-DELTA PRIMEIRA ORDEM

### 3.1. Análise Teórica

Inicialmente foi feita a análise teórica de um modulador Sigma-Delta de primeira ordem. O diagrama de blocos deste modulador está apresentado na Figura 3.1 e é composto por um integrador *switched-capacitor* com função transferência  $\frac{Z^{-1}}{1-\beta Z^{-1}}$ , um comparador e seu respectivo ganho  $G_c$  onde, à saída do mesmo é adicionado o ruído de quantificação  $\frac{V_{NQ}}{V_r}$ , um bloco que representa a tensão de referência e 2 blocos que representam os coeficientes do modulador. O sinal de entrada é representado por  $V_{in}$ ,  $V_o$  é o sinal à saída do integrador e  $d_{out}$  representa a saída do modulador.

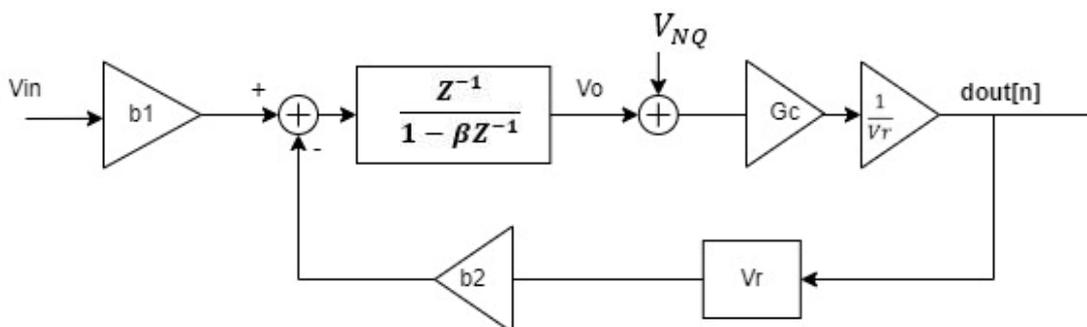


Figura 3.1 - Diagrama de blocos do modulador Sigma-Delta de Primeira ordem, para análise teórica

Começou-se por fazer a análise do integrador *switched-capacitor* partindo da arquitetura da Figura 3.2.

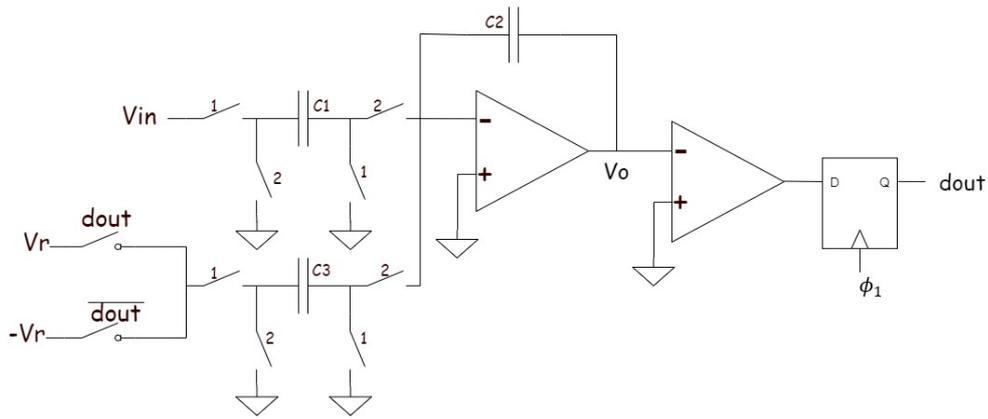


Figura 3.2 - Diagrama do integrador *Switched Capacitor*

No diagrama os interruptores referentes à fase 1,  $\phi_1$ , estão representados com o número 1 e os interruptores referentes à fase 2,  $\phi_2$ , estão representados com o número 2. De modo a realizar a análise do integrador é necessário aplicar a lei da conservação de cargas ( $Q = C * V$ ) e para auxiliar a obtenção das expressões resultantes da aplicação da lei foi obtido um esquema, em que se assume que os interruptores 1 estão ligados, Figura 3.3a, e um esquema assumindo que são os interruptores 2 que estão ligados, Figura 3.3b.

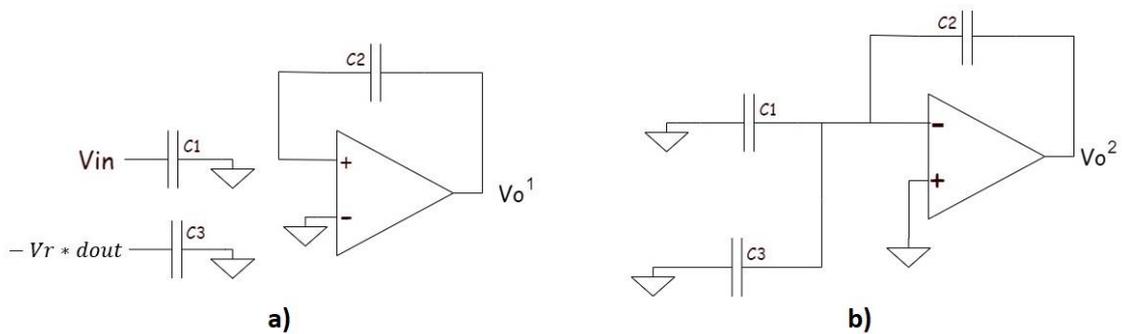


Figura 3.3 - a) Circuito correspondente à  $\phi_1$ ; b) Circuito correspondente à  $\phi_2$

As primeiras equações é no momento de  $\phi_1$  para  $\phi_2$ , ou seja, do instante de amostragem  $[n - 1] * T$  para  $[n - \frac{1}{2}] * T$ . A expressão (3.1) caracteriza a conservação de cargas neste instante.

$$Q_{C_1}^1 + Q_{C_2}^1 + Q_{C_3}^1 = Q_{C_1}^2 + Q_{C_2}^2 + Q_{C_3}^2 \quad (3.1)$$

Da aplicação da expressão da carga,  $Q = C * V$ , na expressão (3.1) resulta a equação (3.2), resolvendo esta equação em ordem a  $V_o \left[ n - \frac{1}{2} \right]$  obtém-se a equação (3.3).

$$\begin{aligned} -V_{in}[n-1] * C_1 + V_r * d_{out}[n-1] * C_1 + V_o[n-1] * C_2 &= \\ &= 0 * C_1 + 0 * C_3 + V_o \left[ n - \frac{1}{2} \right] * C_2 \end{aligned} \quad (3.2)$$

$$V_o \left[ n - \frac{1}{2} \right] = V_o[n-1] - V_{in}[n-1] * \frac{C_1}{C_2} + V_r * d_{out}[n-1] * \frac{C_3}{C_2} \quad (3.3)$$

As equações relativas ao instante  $\phi_2$  para  $\phi_1$ , referem-se ao instante de amostragem  $\left[ n - \frac{1}{2} \right] * T$  para  $[n] * T$ . A equação inicial que caracteriza esta conservação é a equação (3.4).

$$Q_{C_2}^2 = Q_{C_2}^1 \quad (3.4)$$

Aplicando a mesma lógica utilizada nos instantes de amostragem anteriores obtém-se a equação (3.5), sendo que neste caso esta será resolvida em ordem a  $V_o[n]$ , resultando assim a equação (3.6).

$$V_o \left[ n - \frac{1}{2} \right] * C_2 = V_o[n] * C_2 \quad (3.5)$$

$$V_o[n] = V_o \left[ n - \frac{1}{2} \right] \quad (3.6)$$

Estando assim obtidas as equações finais da Conservação de Cargas, utilizando a equação (3.3), que representa  $V_o$  no instante  $\left[ n - \frac{1}{2} \right] * T$ , para substituir na equação (3.6) obtemos a equação final, equação (3.7), para  $V_o$  no instante  $[n] * T$ .

$$V_o[n] = V_o[n-1] - V_{in}[n-1] * \frac{C_1}{C_2} + V_r * d_{out}[n-1] * \frac{C_3}{C_2} \quad (3.7)$$

Tendo as equações da conservação de cargas resolvidas em ordem a  $V_o$  no instante  $[n] * T$  é possível aplicar a transformada de Z de modo a obter uma equação da qual resulta o diagrama de blocos da Figura 3.1. Posto isto, fazendo a transformada e colocando  $\frac{Z^{-1}}{1-Z^{-1}}$  em evidência obtém-se a equação (3.8).

$$\begin{aligned} V_o(Z) &= Z^{-1}V_o(Z) - Z^{-1}\frac{C_1}{C_2}V_{in}(Z) + V_r\frac{C_3}{C_2}d_{out}(Z) * Z^{-1} \Leftrightarrow \\ \Leftrightarrow V_o(Z) &= \frac{Z^{-1}\frac{C_1}{C_2}V_{in}(Z) + V_r\frac{C_3}{C_2}d_{out}(Z) * Z^{-1}}{1 - Z^{-1}} \Leftrightarrow \end{aligned} \quad (3.8)$$

$$\Leftrightarrow V_o(Z) = \frac{Z^{-1}}{1 - Z^{-1}} * (V_{in} * \frac{C_1}{C_2} - \frac{C_3}{C_2} * V_r * d_{out})$$

A equação (3.8) representa o sinal à saída do integrador  $V_o$ , e fazendo a relação com a Figura 3.1,  $\frac{C_1}{C_2}$  correspondente ao coeficiente  $b1$  e  $\frac{C_3}{C_2}$  corresponde ao coeficiente  $b2$ , resultando assim a equação (3.9)

$$V_o(Z) = \frac{Z^{-1}}{1 - Z^{-1}} * (V_{in} * b1 - b2 * V_r * d_{out}) \quad (3.9)$$

Resta obter a expressão de  $d_{out}$ , equação (3.10) que representa a saída do modulador na Figura 3.1.

$$d_{out} = \frac{V_{NQ}}{V_r} + V_o(Z) * \frac{1}{V_r} \quad (3.10)$$

Substituindo  $V_o(Z)$  na equação (3.10) pela sua equação (3.9) obtemos a equação  $d_{out}$  em função dos coeficientes, de  $V_r$ , de  $V_{in}$  e também de  $V_{NQ}$  como demonstrado nas equações (3.11).

$$\begin{aligned} d_{out} &= \frac{V_{NQ}}{V_r} + \frac{1}{V_r} * \left( \frac{Z^{-1}}{1 - Z^{-1}} * (V_{in} * b1 - b2 * V_r * d_{out}) \right) \Leftrightarrow \\ \Leftrightarrow d_{out} + d_{out} * V_r * b2 * \frac{1}{V_r} * \frac{Z^{-1}}{1 - Z^{-1}} &= \frac{V_{NQ}}{V_r} + \frac{Z^{-1}}{1 - Z^{-1}} * \frac{1}{V_r} * V_{in} * b1 \Leftrightarrow \\ \Leftrightarrow d_{out} \left( 1 - \frac{Z^{-1}}{1 - Z^{-1}} * b2 * V_r * \frac{1}{V_r} \right) &= \frac{V_{NQ}}{V_r} + \frac{Z^{-1}}{1 - Z^{-1}} * \frac{1}{V_r} * V_{in} * b1 \Leftrightarrow \\ \Leftrightarrow d_{out} &= \frac{\frac{V_{NQ}}{V_r} * (1 - Z^{-1}) + \frac{1}{V_r} * V_{in} * b1 * Z^{-1}}{1 - Z^{-1} + Z^{-1} * b2 * V_r} \end{aligned} \quad (3.11)$$

Considerando que  $b1 = 1$  e  $b2 = 1$  e assumindo também que  $V_r = 1$  para simplificar a expressão, a expressão final de  $d_{out}$  é a expressão (3.12).

$$d_{out} = V_{NQ} * (1 - Z^{-1}) + V_{in} * Z^{-1} \quad (3.12)$$

Através da equação final de  $d_{out}$  podemos obter a STF e a NTF representadas, respetivamente, pelas equações (3.12) e (3.13). Para obter a expressão da NTF assume-se  $V_{in} = 0$  e para STF é assumido  $NQ = 0$ .

$$NTF = \left. \frac{d_{out}}{V_{NQ}} \right|_{V_{in}=0} = 1 - Z^{-1} \quad (3.13)$$

$$STF = \left. \frac{d_{out}}{V_{in}} \right|_{V_{Nq}=0} = Z^{-1} \quad (3.14)$$

## 3.2. Funcionamento do Modulador de 1ª Ordem e Filtros

De modo a entender o funcionamento do modulador Sigma-Delta Incremental de Primeira Ordem, neste subcapítulo será simulado o Modulador Sigma-Delta de primeira ordem e posteriormente será também explicado o funcionamento de duas arquiteturas distintas, para a decodificação do sinal à saída do modulador.

Para simular o Modulador será tido em consideração do diagrama de blocos da Figura 3.4. Os blocos presentes no diagrama representam o coeficiente de ganho,  $b1$ , o integrador ao qual é feito reinicialização, o comparador e tensão de referência. Os sinais  $V_{in}$ ,  $x_2$  e  $d_{out}[n]$  representam o sinal de entrada, a saída do integrador e a saída do modulador.

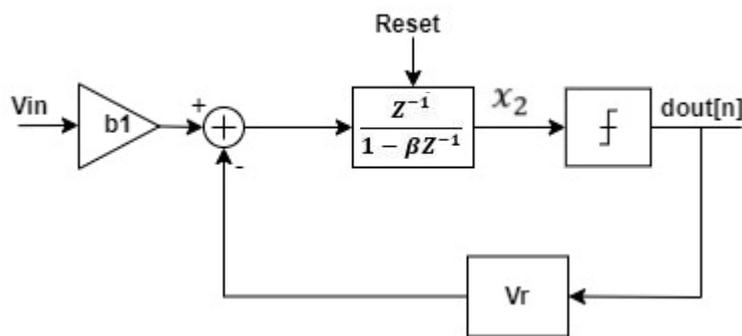


Figura 3.4 - Diagrama de blocos do modulador Sigma-Delta Incremental de Primeira ordem

O integrador presente na arquitetura é um integrador de tempo discreto, o qual é reinicializado no final de cada amostragem de modo a não haver memória de amostragens anteriores. Este integrador, que tem o seu circuito apresentado na Figura 3.5 é do tipo *Switched-Capacitor*, sendo um integrador que se distingue em duas fases, a primeira fase, quando  $\phi_1$  está ativo em que o sinal é amostrado no condensador e a segunda fase quando  $\phi_2$  está ativo é quando é feita a integração da carga acumulada em  $C_1$ , que foi transferida para o condensador  $C_2$ . À saída do integrador vai sendo alterada a tensão de saída consoante a integração da tensão de entrada. Ao fim de OSR amostras o condensador  $C_2$  será curto-circuitado, tornando a tensão de saída do integrador igual a zero e eliminando a memória de sinais de entrada anteriores. Em paralelo com o condensador

$C_2$  encontra-se o interruptor responsável pela reinicialização do sistema. De realçar que o sinal alimentado para o integrador não é apenas  $V_{in}$  mas sim a diferença entre  $V_{in}$  e o sinal referente à amostra anterior, proveniente da saída do comparador.

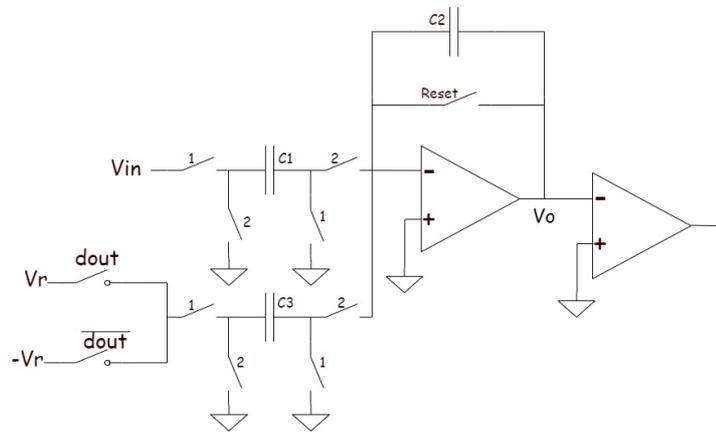


Figura 3.5 – Integrador Switched Capacitor com realimentação

O comparador localizado à saída do modulador é o elemento que introduz o ruído de quantificação,  $V_{NQ}$ , no sistema. É este componente que gera o sinal  $d_{out}$ , comparando os valores da saída do integrador,  $x_2$ , com 0, seguindo o algoritmo da expressão (3.15).

$$d_{out}[n] = \begin{cases} 1, & \text{se } x_2[n] > 0 \\ 0, & \text{se } x_2[n] = 0 \\ -1, & \text{se } x_2[n] < 0 \end{cases} \quad (3.15)$$

É este sinal,  $d_{out}$ , que realimenta o sistema realizando a diferença com  $V_{in}$  e essa diferença leva a que a saída do modulador,  $d_{out}$  guarde a média do sinal de entrada,  $V_{in}$ , como se pode verificar no gráfico da Figura 3.6, onde se pode ver a semelhança entre a média da entrada e da saída para diferentes valores de  $A_{in}$ , a amplitude de entrada.

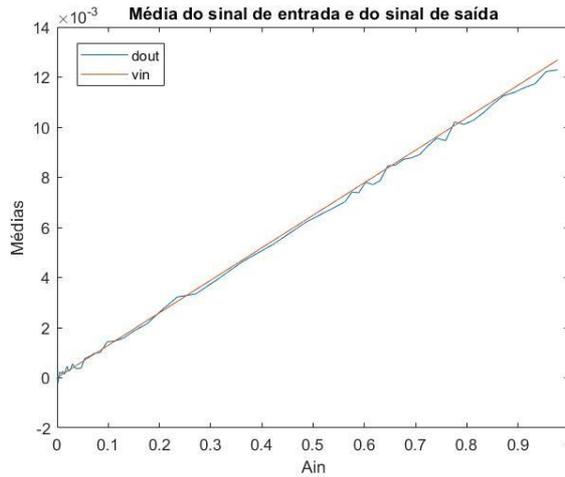


Figura 3.6 – Gráfico de comparação da média do sinal de entrada e do sinal de saída em função da amplitude de entrada,  $OSR = 1200$ ,  $Nmedias = OSR$

### 3.2.1. Implementação com Filtro Col

O Modulador Sigma-Delta necessita de um filtro decimador, filtro digital para decodificar o seu sinal de saída, como tal uma das opções que poderia ser aplicada à saída do modulador é um filtro Sinc. Este filtro tem como objetivo gerar as médias do sinal  $d_{out}[n]$ , seguindo a expressão (3.16).

$$medias = \frac{\sum_{i=0}^{Nmedias-1} dout(i)}{Nmedias} \tag{3.16}$$

A representação desta expressão é o circuito da Figura 3.7, o sinal é somado ao longo de  $N$  médias permitindo que exista um valor de médias para cada *clock*.

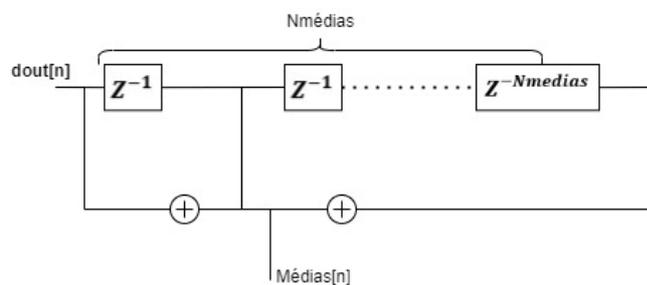


Figura 3.7 – Filtro Sinc

Como se pode observar para realizar este filtro seria necessário o sinal entrar em  $N$  médias de memória o que obrigaria a uma presença de  $N$  somadores, o que torna a

arquitetura muito pesada em termos *hardware*. De modo a solucionar esse problema, obtém-se a função transferência do circuito da Figura 3.7, expressão (3.17).

$$H(Z) = 1 + Z^{-1} + Z^{-2} + \dots + Z^{-Nmédias+1} \quad (3.17)$$

Da conjugação da função transferência com a expressão das médias (3.16) resulta a função transferência (3.18).

$$H(Z) = \frac{1 - Z^{-Nmédias}}{1 - Z^{-1}} \quad (3.18)$$

A esta função transferência corresponde o circuito da Figura 3.8 onde temos um integrador *forward difference*, com atraso na saída do integrador que está ligada a um diferenciador que começa a produzir as amostras ao fim de  $N$  médias.

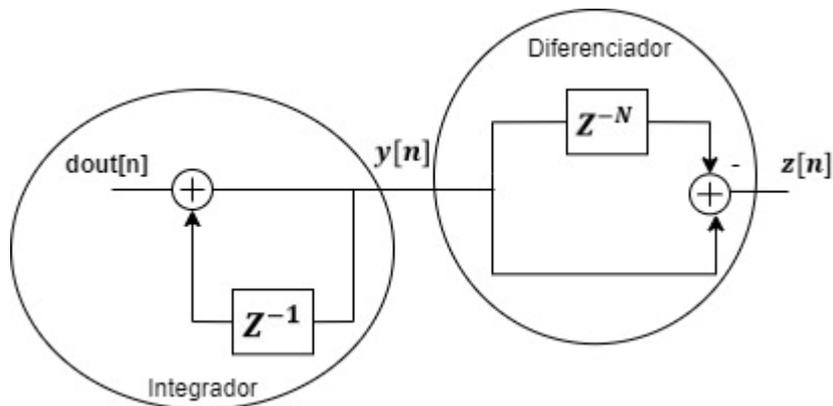


Figura 3.8 - Integrador e diferenciador do Filtro Digital

As saídas  $y[n]$  e  $z[n]$  são descritas pelas expressões (3.19) e (3.20), respetivamente.

$$y[n] = y[n - 1] + d_{out}[n] \quad (3.19)$$

$$z[n] = y[n] - y[n - Nmédias] \quad (3.20)$$

Este circuito já implica menos *hardware* pois já não é necessário implementar a cascata de somadores que seria necessário anteriormente. No entanto não deixa de ser necessário  $N$  médias de memória para o bloco do diferenciador, para solucionar isso será feita uma sub-amostragem à entrada do diferenciador, o que permite que o diferenciador apenas possua um atraso e realize a diferença entre o sinal atrasado e original. Na Figura 3.9 estão visíveis as alterações descritas.

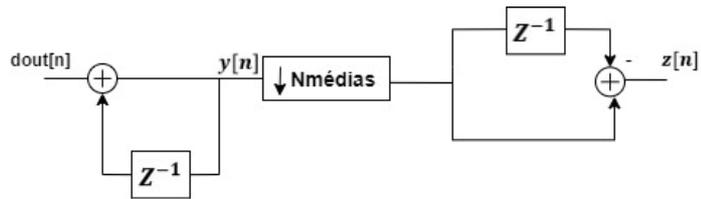


Figura 3.9 - Filtro com o diferenciador simplificado

A expressão da saída  $z[n]$  é contabilizada para quando aplicada a um sinal de entrada que varia e tem uma duração infinita. Neste caso o filtro Sinc implementa uma média deslizante da saída do modulador. No caso de o sinal de entrada do modulador ser DC e o objectivo ser obter uma média de  $N$  médias amostras da saída do modulador, é possível simplificar o circuito anterior eliminando os diferenciadores. Porque neste caso não é necessário obter uma média deslizante do sinal de saída do modulador, sendo simplesmente necessário garantir que a memória do filtro é inicializada a zero antes de cada cálculo da média. Neste caso as médias podem ser obtidas diretamente pelo integrador pois se contarmos  $N$  médias no integrador obtemos, no fim, o valor correto das médias.

Posto isto este filtro foi adicionado ao Modulador Sigma-Delta, como se pode observar na Figura 3.10, de modo a podermos posteriormente analisar o modulador, e o filtro. Este filtro é conhecido com Col (*Cascade of Integrators*), sendo que para o caso do modulador de primeira ordem são dois integradores em cascata, sendo assim um filtro Col de segunda ordem.

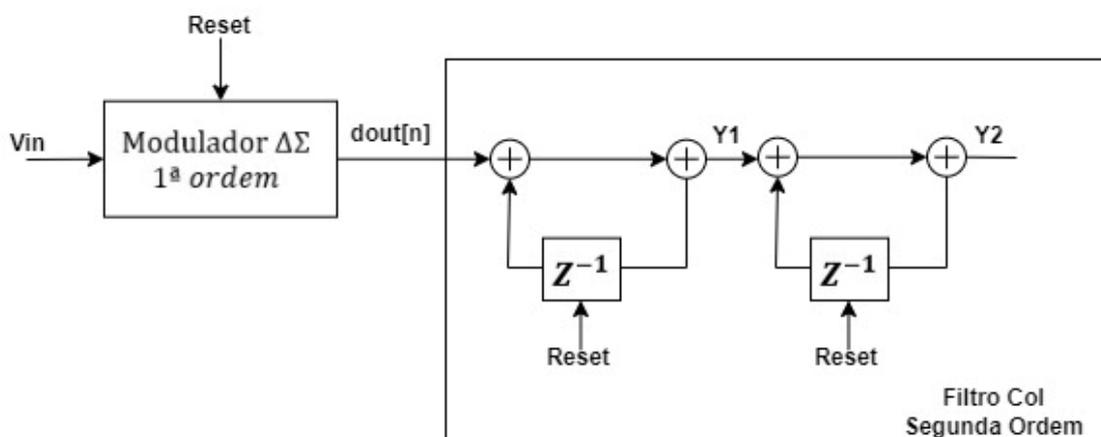


Figura 3.10 - Diagrama de blocos do modulador de primeira ordem implementado com filtro Col de segunda ordem

No diagrama da figura estão identificados os sinais de saída do primeiro e do segundo integrador, Y1 e Y2, respetivamente. Para demonstrar o que foi explicado anteriormente é mostrado nos gráficos das figuras 3.11 e Figura 3.12 os sinais de saída onde se pode notar que o sinal é amostrado durante  $N$  médias e posteriormente é reinicializado de modo que o integrador pare de acumular, e faça a amostragem seguinte.

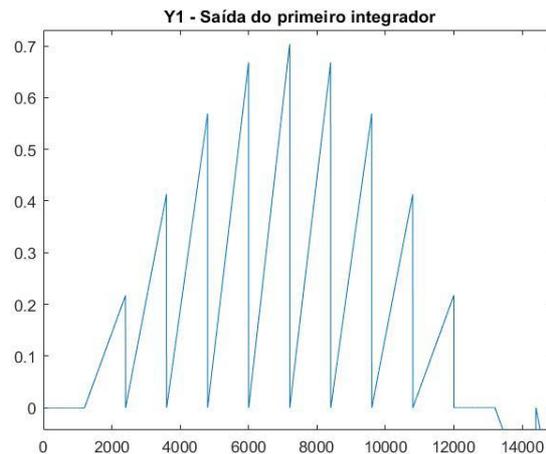


Figura 3.11 - Gráfico da saída do primeiro integrador do filtro Col em função das amostras

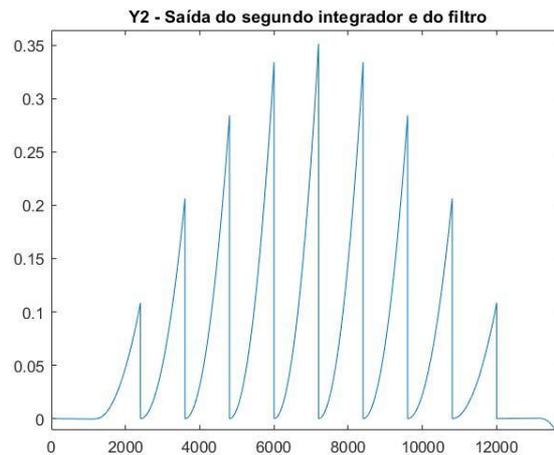


Figura 3.12 - Gráfico da saída do segundo integrador do filtro Col em função das amostras

### 3.2.2. Implementação com Filtro Ótimo

Nesta secção será explicado o funcionamento de um outro filtro que pode ser utilizado para decodificar o sinal recebido do Modulador Sigma-Delta Incremental de Primeira

ordem, o qual é chamado de filtro ótimo[27] e tem o seu respetivo diagrama de blocos representado na Figura 3.13. O modulador é composto por um integrador com função transferência  $Z^{-1}$ , e pelo comparador à saída, sendo por fim, a sequência à saída do modulador decodificada pelo filtro ótimo. De notar que tanto o integrador do modulador como o filtro ótimo são reinicializados no início de cada amostragem.

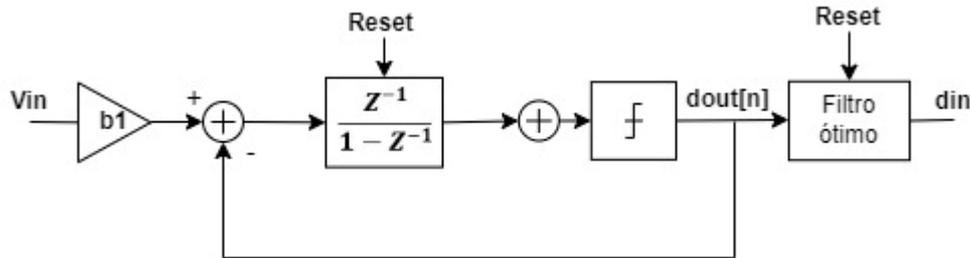


Figura 3.13 - Diagrama de blocos do Modulador Sigma-Delta de 1ª ordem implementado com Filtro Ótimo

A decodificação ótima segue um algoritmo que permite progressivamente restringir os limites do valor de  $V_{in}$ , com isso será possível encontrar um limite inferior máximo,  $lb_{max}$  e o um limite superior mínimo,  $ub_{min}$ . De modo a encontrar os valores candidatos a limites inferior e superior é seguido um algoritmo de decodificação ótima. Para entender este algoritmo é necessário ter em mente que a saída do modulador  $d_{out}[n] \in \{-1,0,1\}$ ,  $S_n$  representa a soma de  $d_{out}[0]$  até  $d_{out}[n-1]$  e  $n$  representa o número de ciclo de quantização. Posto isto, o algoritmo é descrito por 3.21.

$$\begin{cases} lb[n] = \frac{Sn}{n}, se d_{out}[n] \geq 0 \wedge \frac{Sn}{n} \geq lb[n] \\ ub[n] = \frac{Sn}{n}, se d_{out}[n] < 0 \wedge \frac{Sn}{n} \leq ub[n] \end{cases} \quad (3.21)$$

Como se pode ver na definição do algoritmo a cada novo ciclo é definido um novo limite de temporário, ou seja, a cada ciclo de quantização vai ser definido um novo limite de forma que o sinal de entrada seja limitado ao máximo por  $S_n$ . No entanto por vezes pode não ser possível aplicar este algoritmo pois com a entrada do ruído térmico o valor de  $V_{in}$  poderá não estar inserido nos limites. A decodificação ótima sugerida em [27] baseia-se no algoritmo de decodificação ótima anteriormente definido, mas com o objetivo de limitar o ruído térmico e ter alta eficiência energética. Sendo assim o algoritmo de decodificação proposto pode então ser dividido em dois casos:

- $lb_{max} \leq ub_{min}$

Neste caso os limites ainda conseguem definir uma região onde se possa situar a estimativa de  $V_{in}$ ,  $V_{in_{estimado}}$  sendo esta definida por (3.22)

$$vin_{estimado} = \frac{lb_{max} + ub_{min}}{2} \quad (3.22)$$

- $lb_{max} > ub_{min}$

Este é cenário em que os limites não definem uma região para  $vin_{estimado}$ , neste caso é utilizado o número de ciclos pelos quais passou cada um dos limites, considerando  $p_{lb}$  o número de ciclos relativos ao limite inferior e  $p_{ub}$  os ciclos relativos ao limite superior.  $V_{in_{estimado}}$  é então definido por (3.23).

$$V_{in_{estimado}} = \begin{cases} lb_{max}, & \text{se } p_{lb} > p_{ub} \\ ub_{min}, & \text{se } p_{lb} < p_{ub} \end{cases} \quad (3.23)$$

Por fim para mostrar este algoritmo é exposto o gráfico, Figura 3.14, referente ao  $V_{in_{estimado}}$  que corresponde ao sinal  $din$  que se localiza à saída do conversor na Figura 3.13.

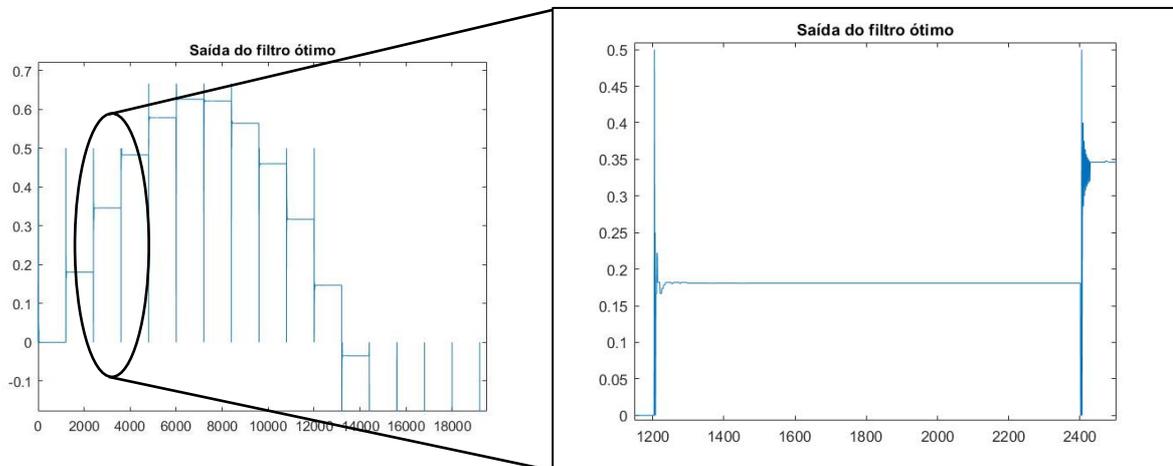


Figura 3.14 - Gráfico da saída filtro ótimo em função das amostras

Ao observar o sinal de saída do filtro ótimo verifica-se que o filtro faz estimativas definindo limites a cada ciclo, como se pode verificar pela linha superior ao valor no início da amostragem que define o limite máximo, aproximando-se assim do valor amostrado, sendo esta amostragem a uma taxa de OSR. No fim do ciclo de amostragem é feita a reinicialização do filtro como se pode ver na figura destacada.

### 3.3. Análise de alto nível

#### 3.3.1. Ambiente de simulação

Começando então por explicar o sinal de entrada, referido sinal  $V_{in}$ , é caracterizado por uma senoide que oscila entre  $\pm A_{in}$ , amplitude de entrada e com frequência,  $F_{sinal} = 0,4 * 10^3 \text{ Hz}$ . Esta senoide é sujeita a um processo de S&H originando o sinal definido pelo gráfico da Figura 3.15

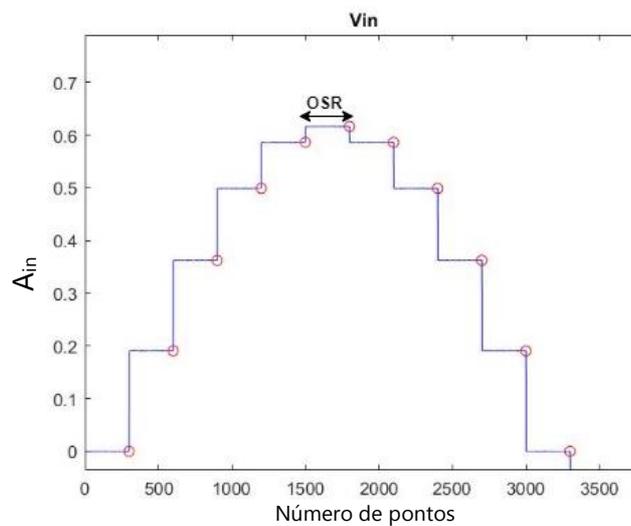


Figura 3.15 - Sinal de entrada  $V_{in}$  com taxa de amostragem OSR

O sinal tem uma frequência de amostragem que resulta da conjugação do OSR, do modulador, com o  $OSR_{sinal}$ , OSR do sinal. O OSR do modulador representa o número de amostras,  $N$ , e o  $OSR_{sinal}$  consiste na razão entre a frequência de amostragem e a frequência do sinal de entrada. As  $N$  amostras passam por um processo de decimação e assim resulta apenas 1 amostra que corresponde a uma senoide de  $n$  pontos por período. Os  $n$  número de pontos são definidos pelo  $OSR_{sinal}$ .

O  $OSR_{sinal}$  pode ser definido por um número inteiro de 10 pontos ou por um número fracionário de pontos próximo de 10. Usando 10 pontos obtém-se sempre o mesmo valor a cada período da sinusoidal, ao contrário do que acontece usando um valor fracionário próximo de 10, permitindo assim obter diferentes pontos a cada período.

Na Figura 3.16 e na Figura 3.17 estão representados, respetivamente, alguns períodos da senoide utilizando  $OSR_{\text{sin}} = 10$ , onde se pode verificar que os valores que são decimados a cada período são sempre os mesmos e a FFT do mesmo sinal.

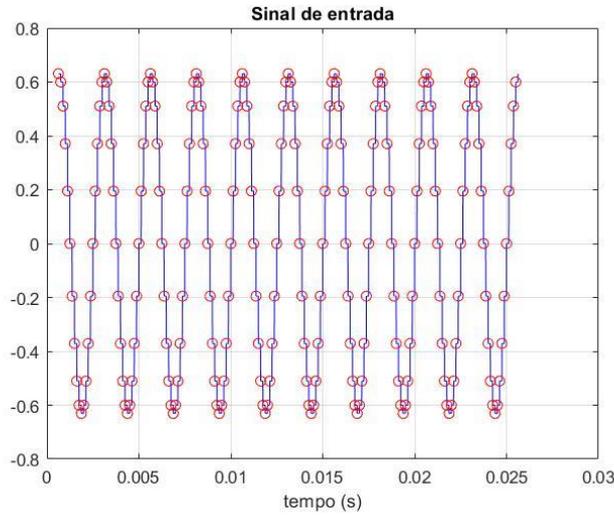


Figura 3.16- Sinal de entrada com  $OSR_{\text{sin}} = 10$

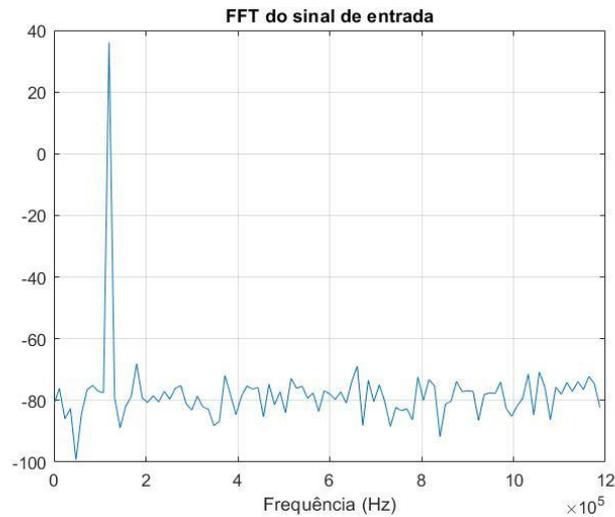


Figura 3.17- FFT do sinal de entrada com  $OSR_{\text{sin}} = 10$

No gráfico da Figura 3.18 observa-se alguns períodos da senoide resultante de  $OSR_{\text{sin}} = \frac{1200}{111}$ , onde se pode ver que a cada período são obtidos diversos pontos da senoide. A FFT deste sinal é mostrada na Figura 3.18

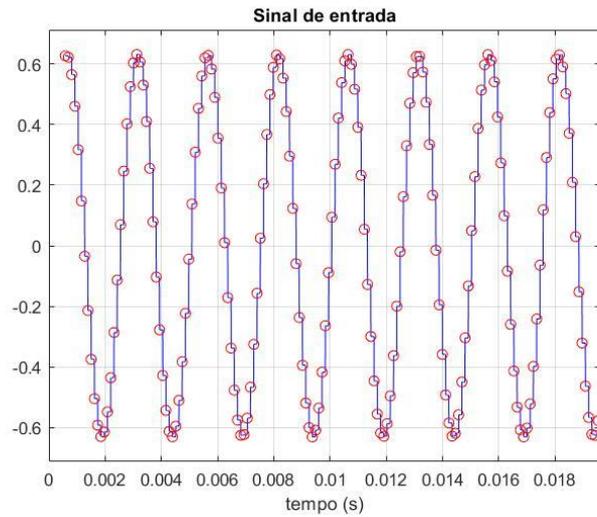


Figura 3.18 - Sinal de entrada com  $OSR_{sinal} = \frac{1200}{111}$

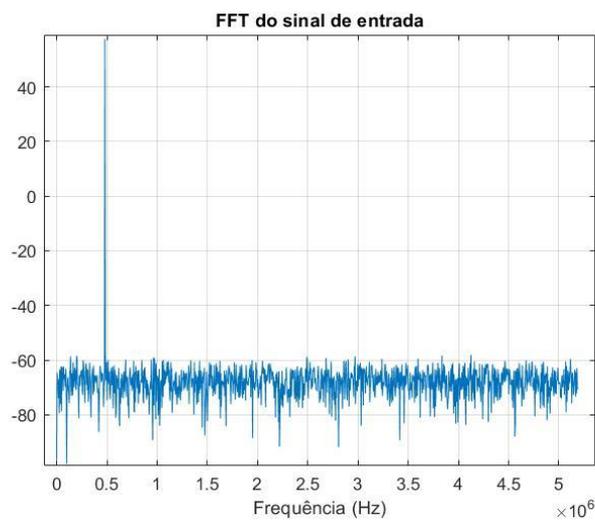


Figura 3.19 - FFT do sinal de entrada com  $OSR_{sinal} = \frac{1200}{111}$

Ao sinal  $V_{in}$  é ainda adicionado o ruído térmico,  $V_{nt}$ , que provoca distorção na amostragem prejudicando assim a precisão do valor de saída. Este ruído deve-se na sua maioria aos *ampops* presentes no circuito. Na Figura 3.20 está destacada a sua influência no sinal de entrada.

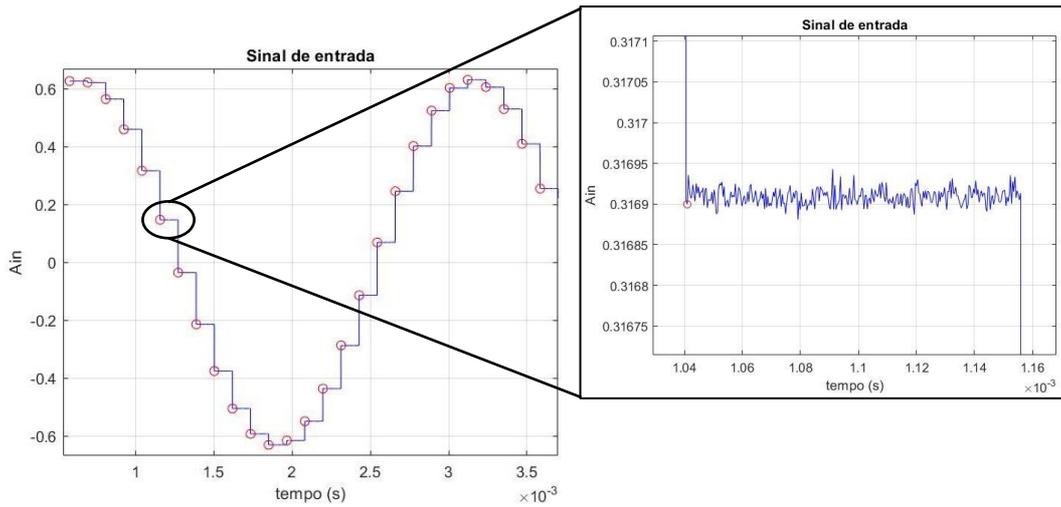


Figura 3.20 - Gráfico do sinal de entrada com pontos amostrados e com influência do ruído térmico destacado

### 3.3.2. Simulações de alto-nível

Neste sub-capítulo as estruturas descritas em 3.2.1 e 3.2.2 são alvo do estudo de alto nível com o auxílio do *software Octave*. Para este estudo foram utilizadas as duas arquiteturas que foram mencionadas na secção anterior, e os diagramas utilizados para a sua modulação estão na Figura 3.13, que corresponde ao modelo do Modulador Sigma-Delta de 1ª ordem implementado com o filtro ótimo e na Figura 3.21, que corresponde ao mesmo modulador mas implementado com o Col de segunda ordem.

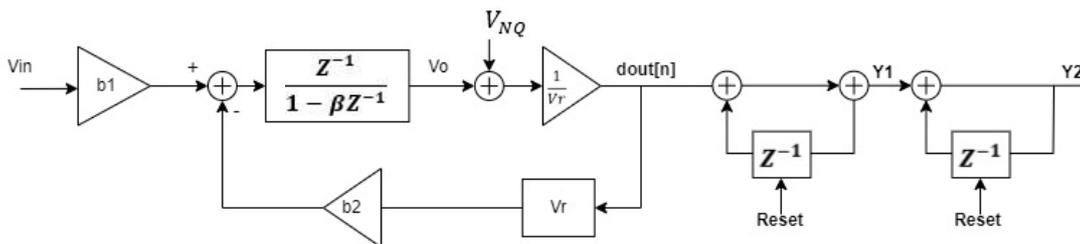


Figura 3.21 – Diagrama do modulador de primeira ordem com filtro Col de segunda ordem

Para obter o valor de  $A_{in}$  é analisado o gráfico da Figura 3.22 de onde se conclui que o valor que maximiza a SNDR do sistema é  $-0.8\text{dB}$ , para ambas as arquiteturas.

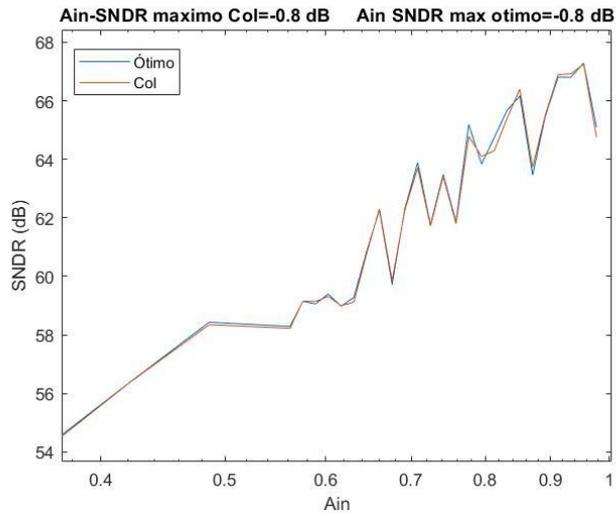


Figura 3.22 - Gráfico de SNDR em função da amplitude de entrada, com ponto máximo de SNDR marcado

Para estes dois conversores, vão ser analisados alguns dos seus parâmetros de forma a otimizar algumas métricas como SNDR ou DR. Esta análise vai ser feita, tendo em conta integradores ideais e reais no modulador Sigma-Delta, sendo que a nível de modulação, esta não idealidade é representada pelo coeficiente beta que é inserido nas expressões dos integradores.

Estando definido o sinal de entrada é necessário realizar simulações de modo a perceber como se pode otimizar duas métricas importantes nestes ADC, a SNDR e o DR. O objetivo passa por tentar ter os valores mais altos possíveis destas métricas de modo que a presença de ruído afete o menos possível o circuito e o DR de modo que seja possível grandes variações da entrada. Sendo que o modulador deste circuito é de primeira ordem, os parâmetros a ter em conta para análise são o OSR, taxa de amostragem, o beta que é coeficiente de idealidade e o ruído térmico acrescentado a  $V_{in}$  como já foi referido.

Inicialmente, olhando para a implementação como filtro Col observou-se se haveria vantagem de utilizar 2 integradores à saída do modulador um integrador e não apenas um. De modo a fazer esta análise foram obtidos os gráficos das figuras 3.23 e 3.24 que representam, respetivamente, a utilização de integradores ideais no modulador,  $\beta = 1$ , e a utilização de integradores reais no modulador,  $\beta < 1$ .

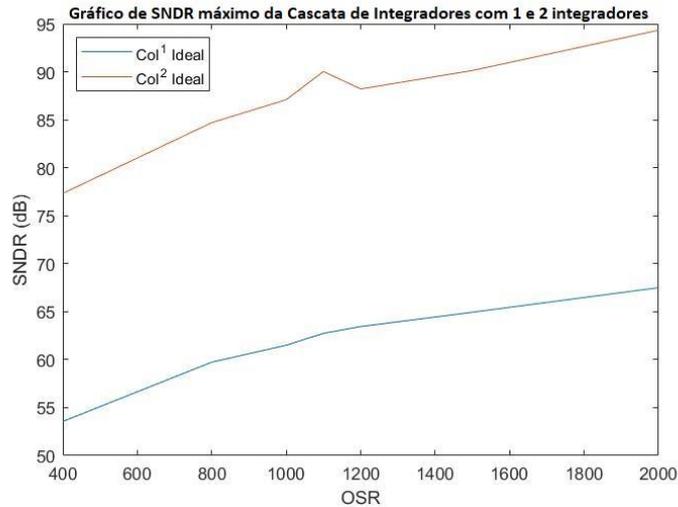


Figura 3.23 - Gráfico de SNDR para as saídas Y1 e Y2 do filtro Col com integradores ideais do modulador reais,  $\beta=1$

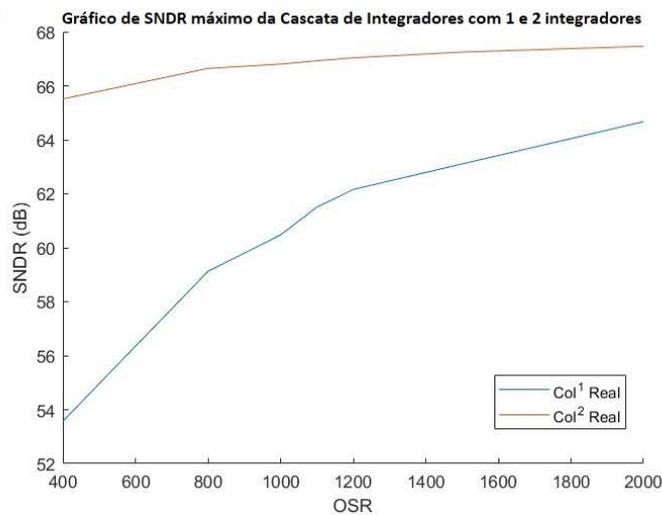


Figura 3.24 - Gráfico de SNDR para as saídas Y1 e Y2 do filtro Col com integradores do modulador reais,  $\beta=0,99$

Analisando estes resultados não fica qualquer dúvida da utilidade que tem acrescentar o segundo integrador, tanto no caso de integradores reais como integradores ideais. Observando por exemplo para um  $OSR=1200$  vemos uma melhoria de cerca de 25 dB quando falamos de integradores ideais, e nos integradores reais uma melhoria de 7 dB. É também de salientar que a saída, Y2, do segundo integrador é bastante mais afetada por  $\beta < 1$  do que a saída do primeiro integrador.

De seguida é comparada a performance de ambas as arquiteturas variando OSR entre 400 e 2000, sendo assim será representado o gráfico de SNDR máximo, Figura 3.25, para o filtro Col de segunda ordem e para o filtro ótimo e é também mostrado o gráfico de DR, Figura 3.26, para os mesmos casos

Col<sup>2</sup>

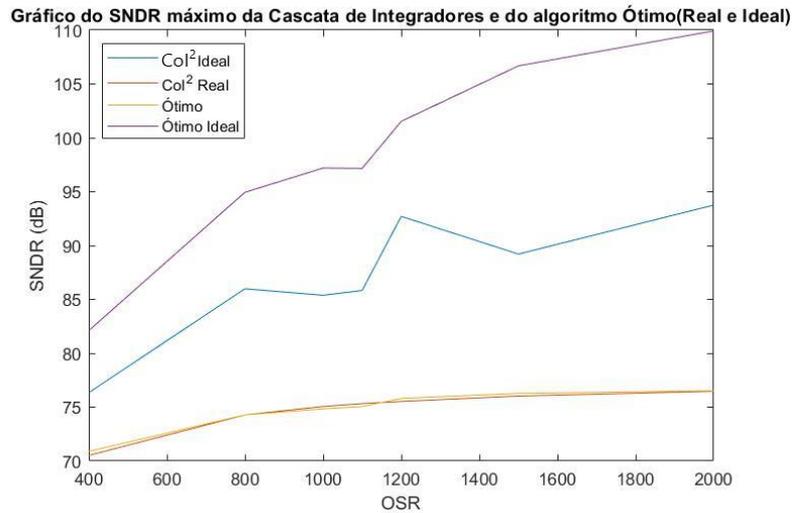


Figura 3.25 - Gráfico do SNDR para a variação de OSR

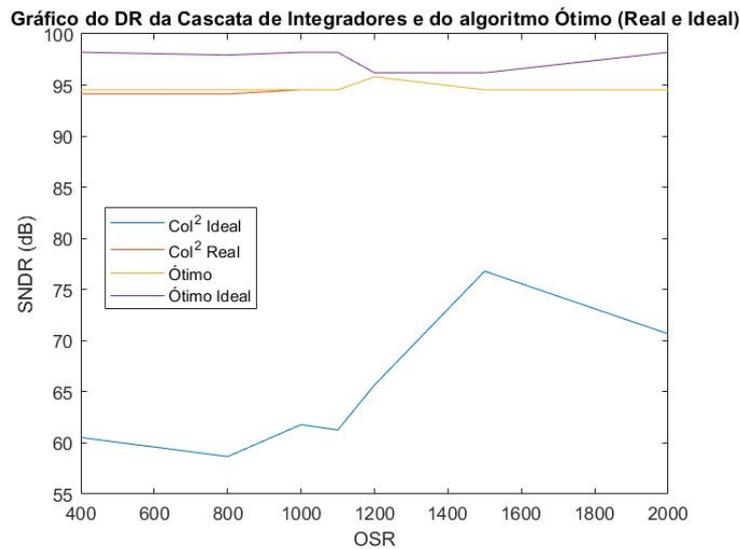


Figura 3.26 - Gráfico do DR para a variação de OSR

Na Figura 3.25 observa-se, como era de esperar, os 2 filtros ideais com uma performance bastante superior aos restantes e destacasse o facto de o filtro ótimo ideal conseguir atingir mais 8 dB de SNDR. Em relação aos dois filtros reais, têm resultados muito idênticos, sendo que neste campo nenhum tem vantagem perante o outro. Falta ainda realçar

que o facto de ser introduzido um valor  $\beta < 1$  afeta bastante mais o filtro ótimo do que o filtro Col. Em relação ao DR os dois filtros reais também se mostram bastante próximos, com cerca de 94 dB e o filtro ótimo ideal cerca de 3 dB superior, o filtro Col ideal obteve um resultado fora do comum o leva a querer que haja alguma falha a nível de modulação. Tentando perceber que valor de OSR pode otimizar o circuito sem prejudicar a velocidade do mesmo poder-se-ia optar por um valor de 1200, que otimiza SNDR para os dois filtros e apesar de o filtro ótimo ideal ter uma pequena quebra no DR com OSR=1200, os filtros reais têm um ligeiro aumento nessa zona.

O próximo parâmetro estudado será o ruído térmico introduzido em  $V_{in}$ , que será alvo de uma variação entre  $0\mu V$  e  $100\mu V$  de modo estudar-se o comportamento de SNDR e DR com essa variação. Nas figuras 3.27 e 3.28 são expostos os resultados oriundos da simulação explicada.

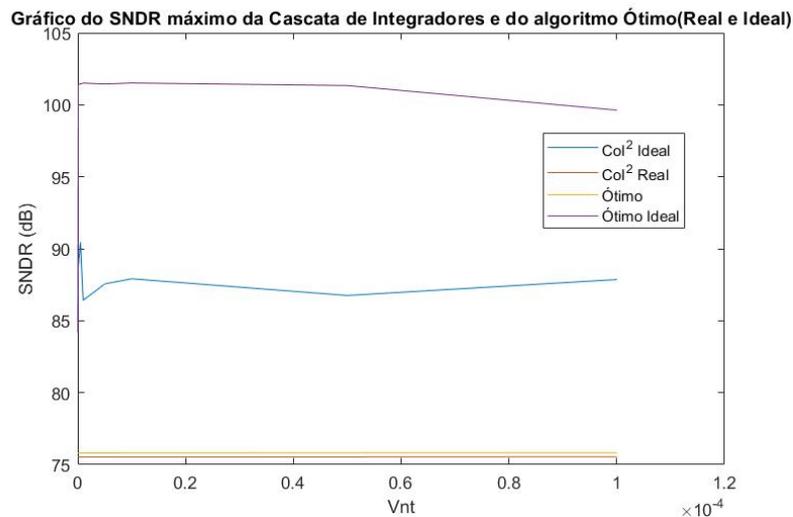


Figura 3.27 - Gráfico do SNDR para a variação de Vnt

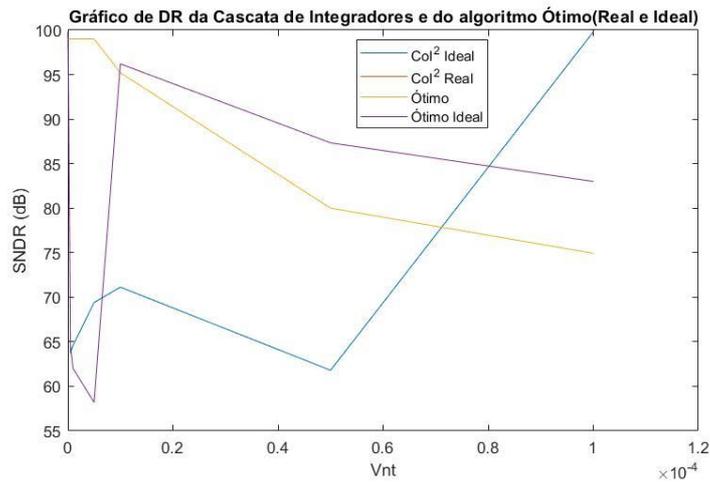


Figura 3.28 -Gráfico do DR para a variação de Vnt

Observando os resultados do comportamento de SNDR com os filtros ideais vê-se que há uma ligeira oscilação para os valores iniciais de Vnt, ruído térmico, o que mostra que para pequenos valores de ruído podem causar uma redução do SNDR, no entanto para valores mais elevados acaba por estabilizar. Em relação aos filtros reais, talvez pela variação que também provoca  $\beta < 1$ , os valores de SNDR são praticamente constantes desde o início. Em relação a DR nota-se que em todos os casos a partir de  $V_{nt} = 10\mu V$  começa a cair o valor de DR. Como tal o valor de  $10\mu V$  é aconselhável.

Por fim realizou-se a análise à variação de beta com valores inferiores a 1 para perceber de que forma esse parâmetro contribuía para a variação de SNDR e de DR. Como tal foram realizadas simulações para valores entre 0.9 e 0.95 e os seus resultados estão nas figuras 3.29 e 3.30.

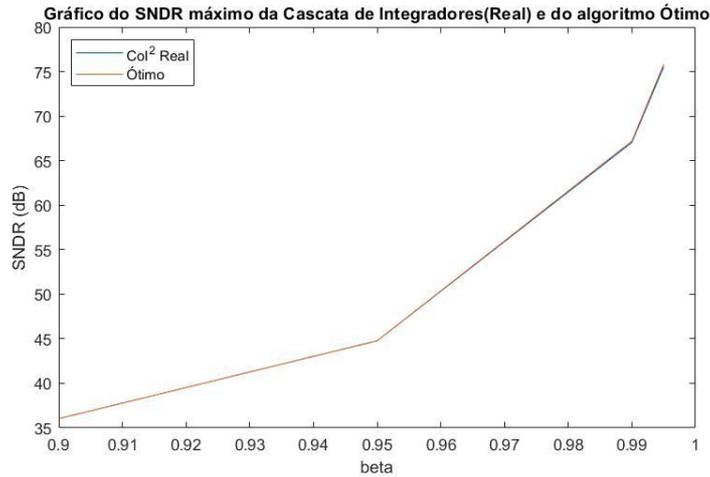


Figura 3.29 - Gráfico do SNDR para a variação do coeficiente beta

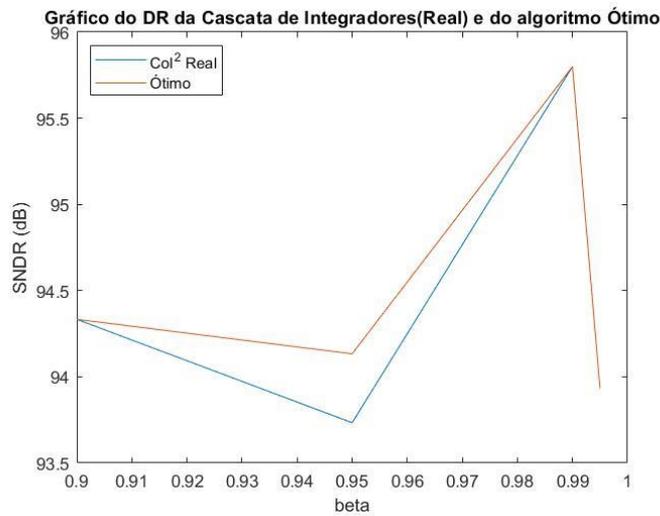


Figura 3.30 - Gráfico do DR para a variação do coeficiente beta

Mais uma vez nestas simulações não houve praticamente nenhuma diferença entre o algoritmo ótimo e o filtro a nível de SNDR, o que se pode verificar é que quanto maior beta maior a SNDR de qualquer um dos sinais, portanto não sendo possível a nível prático obter  $\beta = 1$ , poder-se-ia escolher um valor a rondar os 0,99 de modo a poder otimizar o SNDR também com o valor de beta. Em relação a DR para valores mais baixos de beta o filtro ótimo tem melhor performance, no entanto quando se chega a valores mais elevados de beta, os dois filtros ficam com valores de SNDR idênticos.

Tendo sido realizada esta análise para o modulador de primeira ordem, serão agora mostrados os resultados da SNDR obtida, através dos espectros da FFT, tendo em conta os valores da Tabela 3.1

Tabela 3.1 - Tabela dos parâmetros obtidos através das simulações para modulador de primeira ordem

	Vnt	Beta	OSR	b1	b2
<i>Real</i>	10 $\mu$ V	0,99	1200	1	1
<i>Ideal</i>		1			

Em primeiro lugar é referido o gráfico do espectro de cada saída de cada um dos filtros e respetivas SNDR tendo em conta integradores ideais. No gráfico, Figura 3.31, pode-se confirmar que realmente a junção do primeiro com o segundo integrador, tem um desempenho muito superior do que se tivéssemos apenas o primeiro.

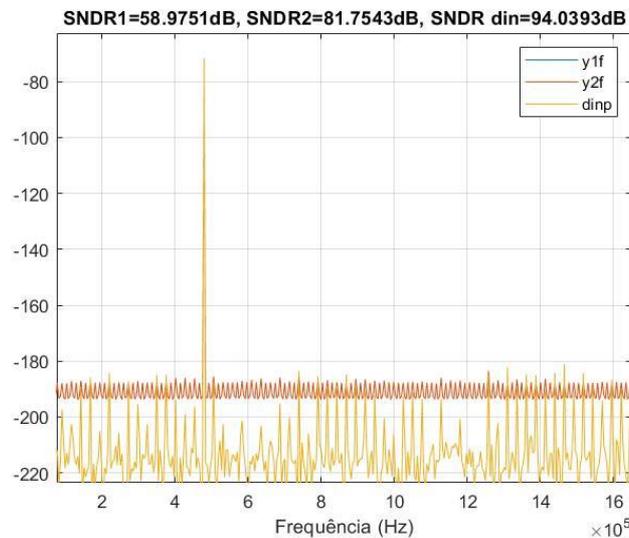


Figura 3.31 - FFT dos sinais de saída com integradores ideais, beta=1

Onde também se tem um grande ganho a nível de SNDR é utilizando o algoritmo ótimo em vez do filtro Col, como já se perspetivava pelas simulações anteriores, para integradores ideais. Em último é mostrado o espectro da FFT das saídas do filtro tendo em conta que estão a ser utilizados integradores reais, com  $\beta = 0,99$ . Esse espectro está definido na Figura 3.32.

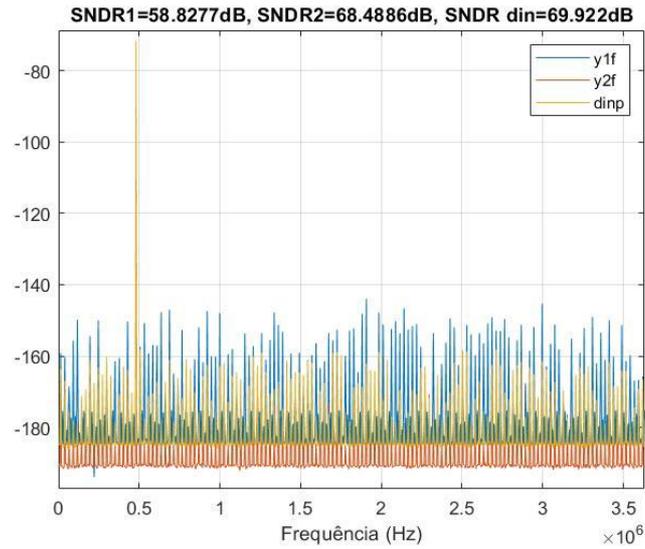


Figura 3.32 - FFT dos sinais de saída com integradores reais,  $\beta=0,99$

Como se pode observar e seguindo o raciocínio já apresentado, a saída do segundo integrador tem um valor de SNDR bastante superior à saída do primeiro integrador o que mostra que se perdia uma grande qualidade na conversão caso não se usasse o filtro Col de segunda ordem. Outra observação a fazer é o facto de apesar de com integradores ideais no modulador o algoritmo ótimo ser bastante superior, quando se introduz não idealidades no circuito o comportamento do filtro ótimo e do filtro Col torna-se idêntico.

## SIGMA-DELTA SEGUNDA ORDEM

### 4.1. Análise Teórica

Neste ponto é feita a análise teórica referente ao modulador Sigma-Delta de Segunda Ordem, da Figura 4.1. Como se pode ver pelo diagrama de blocos é composto por dois integradores *Switched-Capacitor* de tempo discreto com função transferência  $\frac{z^{-1}}{1-\beta z^{-1}}$ , é adicionado o ruído térmico de quantização,  $V_{NQ}$ , e a multiplicação por  $\frac{1}{V_r}$ , de modo a obter  $NQ$ , notando que estes dois blocos correspondem ao comparador. Fazem ainda parte do modulador os blocos dos coeficientes e da tensão de referência. No diagrama de blocos está ainda referenciado o sinal de entrada  $V_{in}$ , o sinal  $V_o^1$  que representa a saída do primeiro integrador,  $V_o^2$  referente à saída do segundo integrador e  $d_{out}$  que representa a saída do integrador após a comparação.

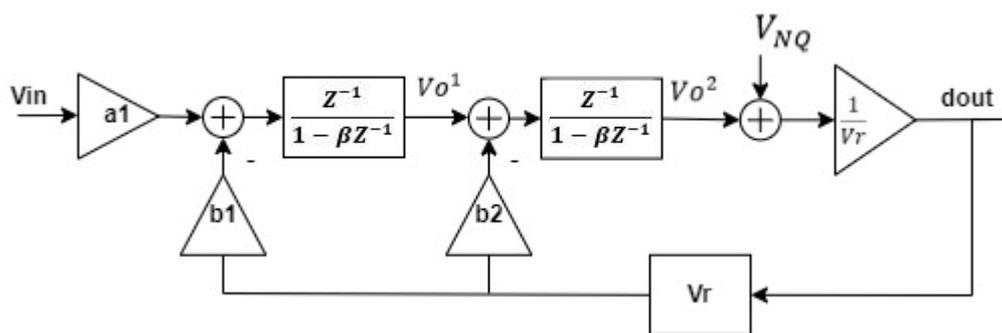


Figura 4.1 - Diagrama de blocos do modulador Sigma-Delta de Segunda ordem, para análise teórica

Pela análise do modulador Sigma-Delta de Primeira ordem temos que  $V_o^1$  tem a expressão (4.1) já obtida no capítulo 3.1.

$$V_o^1(Z) = \frac{Z^{-1}}{1 - Z^{-1}} * (V_{in} * b1 - b2 * V_r * d_{out}) \quad (4.1)$$

Pela análise do diagrama de blocos obtém-se a equação (4.2), para a saída do segundo integrador,  $V_o^2$  em função de  $V_o^1$ .

$$V_o^2 = \frac{Z^{-1}}{1 - Z^{-1}} (-b2 * V_r + V_o^1) \quad (4.2)$$

Substituindo  $V_o^1$ , obtém-se a equação (4.3), que é a equação desenvolvida para a saída do segundo integrador.

$$V_o^2 = \frac{Z^{-1}}{1 - Z^{-1}} \left( -b2 * V_r * d_{out} + \frac{Z^{-1}}{1 - Z^{-1}} (V_{in} * a1 - b1 * V_r * d_{out}) \right) \quad (4.3)$$

Analisando novamente o diagrama obtemos também a equação (4.4), que representa  $d_{out}$  em função de  $V_r$ ,  $V_{NQ}$  e  $V_o^2$ .

$$d_{out} = \frac{V_{NQ}}{V_r} + \frac{1}{V_r} * V_o^2 \quad (4.4)$$

Agora substituindo  $V_o^2$ , na equação (4.4), pela expressão de  $V_o^2$ , da equação (4.3) é possível obter a expressão de  $d_{out}$  em função dos vários blocos e sinais, equação (4.5).

$$d_{out} = \frac{1}{V_r} \left( V_{NQ} + \frac{Z^{-1}}{1 - Z^{-1}} \left( -b2 * V_r * d_{out} + \frac{Z^{-1}}{1 - Z^{-1}} (V_{in} * a1 - b1 * V_r * d_{out}) \right) \right) \quad (4.5)$$

Resolvendo a equação (4.5) em ordem a  $d_{out}$  obtém-se a expressão do sinal à saída do modulador, em função de  $D(Z)$ . Na equação (4.6) é feito o desenvolvimento de  $d_{out}$ , sendo  $a1 = 1$ .

$$\begin{aligned} d_{out} &= \frac{V_{NQ}}{V_r} + \frac{Z^{-2}}{(1 - Z^{-1})^2} * \frac{V_{in}}{V_r} a1 - \left( b2 \frac{Z^{-1}}{1 - Z^{-1}} + b1 \frac{Z^{-2}}{(1 - Z^{-1})^2} \right) * d_{out} \Leftrightarrow \\ &\Leftrightarrow d_{out} * ((1 - Z^{-1})^2 + b1 * Z^{-2} + b2 * Z^{-1} * (1 - Z^{-1})) = \\ &= \frac{V_{NQ}}{V_r} * (1 - Z^{-1})^2 + Z^{-2} * \frac{V_{in}}{V_r} \Leftrightarrow \\ &\Leftrightarrow d_{out} = \frac{V_{NQ}}{V_r} * \frac{(1 - Z^{-1})^2}{D(Z)} + \frac{Z^{-2}}{D(Z)} * \frac{V_{in}}{V_r} \end{aligned} \quad (4.6)$$

Na expressão de  $d_{out}$ , em (4.6) considera-se  $D(Z)$  a expressão (4.7).

$$D(Z) = 1 - (2 - b_2) * Z^{-1} + Z^{-2}(1 + b_1 - b_2) \quad (4.7)$$

Considerando,

$$\begin{cases} b_1 = 1 \\ b_2 = 2 \end{cases} \Rightarrow D(Z) = 1 \quad (4.8)$$

Assim sendo, a expressão de  $d_{out}$ , a expressão de saída do modulador, considerado ideal é a expressão (4.9).

$$d_{out} = \frac{V_{NQ}}{V_r} * (1 - Z^{-1})^2 + \frac{V_{in}}{V_r} * Z^{-2} \quad (4.9)$$

Para simplificar aplica-se  $V_r = 1$  e obtém-se a equação (4.10)

$$d_{out} = V_{NQ} * (1 - Z^{-1})^2 + V_{in} * Z^{-2} \quad (4.10)$$

De modo a obter a STF e a NTF, expressões (4.11) e (4.12), respetivamente, é necessário considerar  $V_{in} = 0$ , para a NTF e  $V_{NQ} = 0$  para a STF.

$$NTF = \left. \frac{d_{out}}{V_{NQ}} \right|_{V_{in}=0} = (1 - Z^{-1})^2 \quad (4.11)$$

$$d_{out} = V_{NQ} * (1 - Z^{-1})^2 + V_{in} * Z^{-2} \quad (4.12)$$

## 4.2. Funcionamento Modulador de 2ª Ordem e Filtros

Neste sub-capítulo é explicado o funcionamento do modulador Sigma-Delta Incremental de Segunda ordem que pode ser utilizado juntamente com dois filtros digitais distintos em que um utiliza um filtro composto por um Col, idealmente de terceira ordem, para decodificar o sinal do modulador e o segundo utiliza um Filtro Ótimo[27]. Posteriormente ao funcionamento do modulador são explicados os dois filtros e apresentados os gráficos de saída.

O diagrama de blocos da Figura 4.2 caracteriza o modulador que será tido em consideração nesta secção. A sua composição é bastante semelhante à arquitetura estudada para o modulador de 1ª ordem, as diferenças baseiam-se na introdução de um segundo integrador com a mesma função transferência, com este integrador é também introduzido o *feedback path* para o mesmo onde surge o coeficiente  $b_2$ .

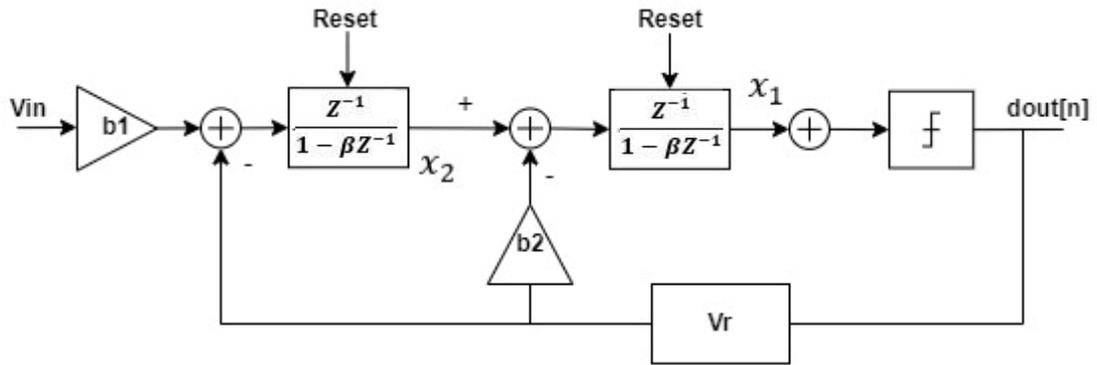


Figura 4.2 - Diagrama de blocos do modulador Sigma-Delta Incremental de Segunda ordem

Os integradores são exatamente os mesmos que os usados na primeira ordem, portanto é um integrador *switched capacitor* passivo operando em tempo discreto, também dividido em 2 operações distintas, primeiro amostragem e depois integração. A realimentação neste caso é feita tanto ao sinal de entrada onde é subtraída pela saída do modulador e a realimentação interna consiste na subtração do sinal de saída, à saída do do primeiro modulador. O comparador comporta-se segundo o algoritmo (4.13) tal e qual como no de primeira ordem.

$$d_{out}[n] = \begin{cases} 1, & \text{se } x_2[n] > 0 \\ 0, & \text{se } x_2[n] = 0 \\ -1, & \text{se } x_2[n] < 0 \end{cases} \quad (4.13)$$

Para o modulador de segunda ordem verifica-se a mesma característica referida no de primeira ordem, simulando a média da entrada do modulador e da saída os valores são bastante idênticos, apenas com ligeiros desvios, mas mais acentuados quando comparado com o modulador de primeira ordem, como se pode ver no gráfico da Figura 4.3.

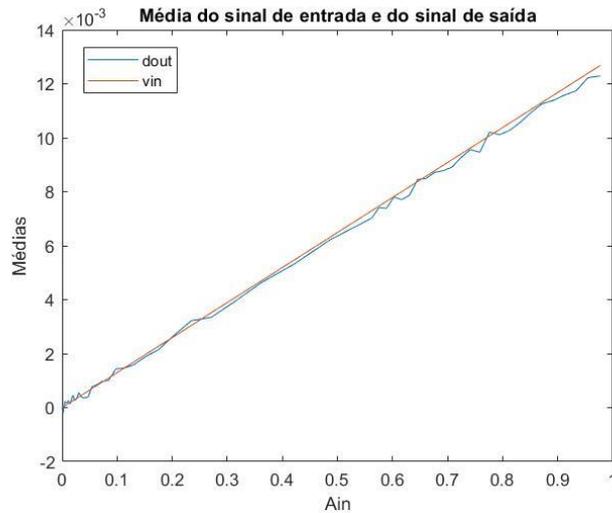


Figura 4.3 - Gráfico de comparação da média do sinal de entrada e do sinal de saída em função da amplitude de entrada para o modulador de segunda ordem

### 4.2.1. Implementação com Filtro Col

O primeiro filtro abordado para decodificar a saída do modulador descrito anteriormente é um filtro Col, Figura 4.4, idêntico ao filtro usado no modulador de primeira ordem sendo a única diferença a adição de um terceiro bloco integrador, criando assim um filtro Col de terceira ordem. Este terceiro bloco foi incluído pois idealmente seria de esperar que ao inserir este terceiro integrador fosse possível obter médias mais precisas e aumentar o valor de SNDR. Como será mostrado mais à frente esta característica não tem o mesmo comportamento quando utilizados integradores reais. É importante referir que todos os integradores são reinicializados no fim de cada amostra.

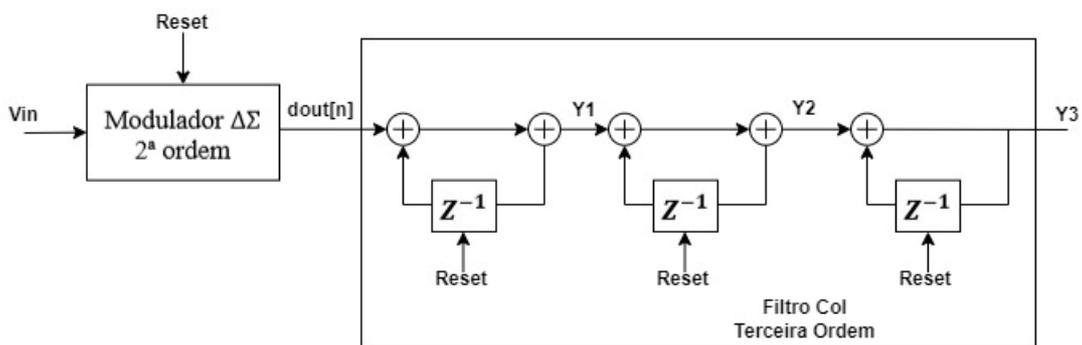


Figura 4.4 - Diagrama de blocos do Modulador Sigma-Delta de 2ª ordem implementado com Filtro Col de 3ª ordem

Os sinais de saída de cada um dos integradores pertencentes ao filtro digital são identificados na Figura 4.4 por Y1 para a saída do primeiro integrador, Y2 para a saída do terceiro integrador e Y3 para a saída do terceiro integrador. O objetivo é que a cada nova média calculada o sinal de saída se aproxime o mais depressa possível do valor de entrada. Para mostrar esta característica, observamos as Figura 4.5, Figura 4.6 e Figura 4.7 onde se nota claramente uma melhoria nos resultados obtidos, como se pode ver comparando a saída do primeiro e do terceiro integrador, onde no primeiro existe uma grande distorção ao longo da amostragem. Outra característica bem definida nos gráficos é o facto de ao fim de cada amostragem o integrador ser reinicializado, de modo a não haver qualquer tipo de resíduos para as amostras seguintes.

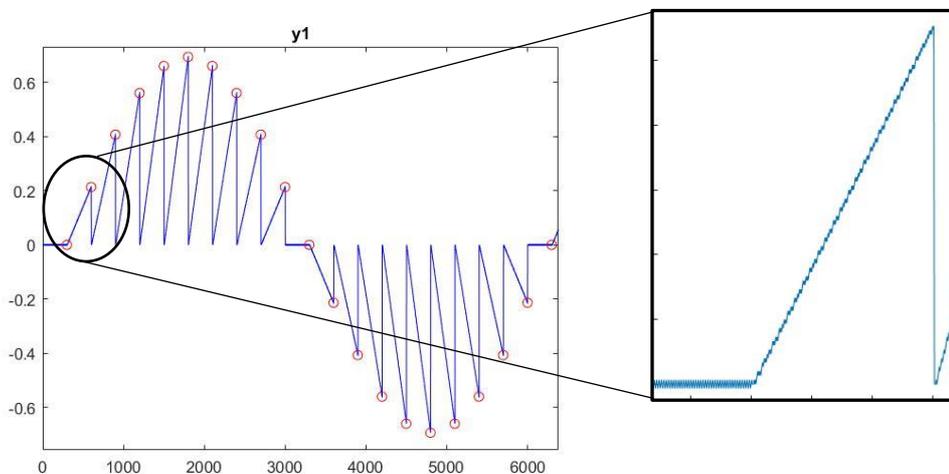


Figura 4.5 - Gráfico da saída do primeiro integrador do filtro Col em função das amostras

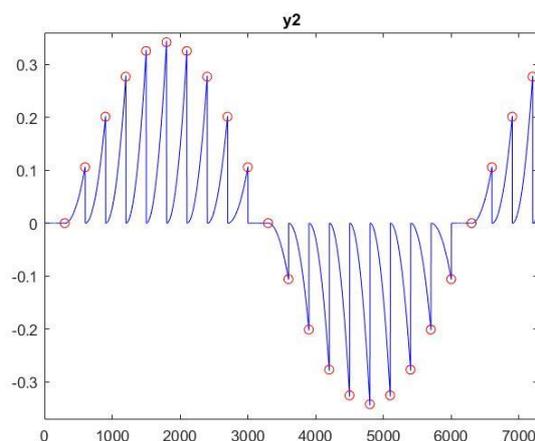


Figura 4.6 - Gráfico da saída do segundo integrador do filtro Col em função das amostras

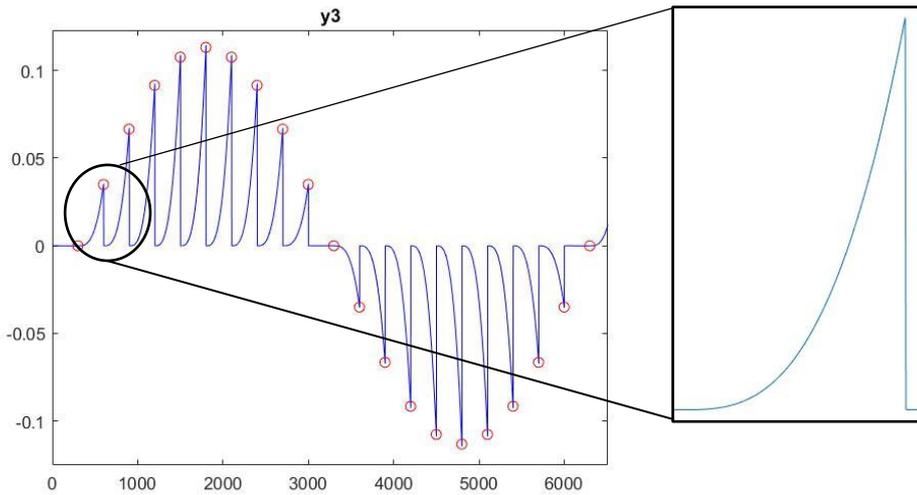


Figura 4.7 - Gráfico da saída do terceiro integrador do filtro Col em função das amostras

## 4.2.2. Implementação com Filtro Ótimo

Na segunda arquitetura, com o diagrama de blocos apresentado na Figura 4.8, baseada em [27], o modulador é composto por dois integradores com função transferência  $Z^{-1}$  sendo feito a reinicialização a ambos os integradores, a saída do modulador é decodificada por um Filtro Ótimo que também é reinicializada.

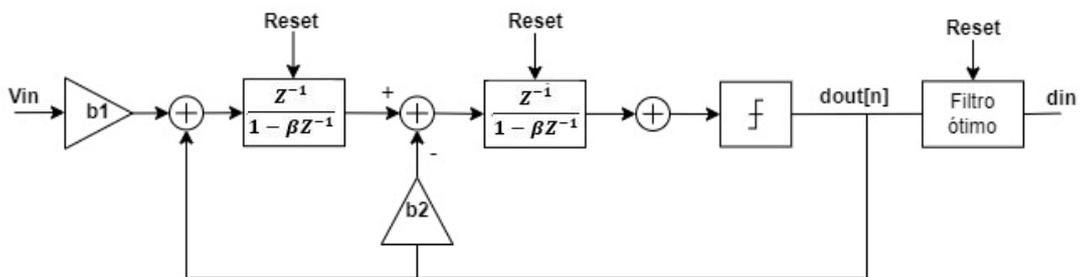


Figura 4.8 - Diagrama de blocos do Modulador Sigma-Delta de 2ª ordem implementado com Filtro Ótimo

O algoritmo que é desenvolvido para implementar o filtro ótimo para o modulador é em tudo semelhante ao implementado para a primeira ordem, sendo a principal diferença a entrada dos coeficientes do circuito na soma,  $S_n$ , que soma as saídas do modulador,  $d_{out}[n]$ . Para comprovar que o algoritmo também é aplicado com este modulador, é exposto o sinal de saída,  $d_{in}$ , do filtro ótimo, Figura 4.9

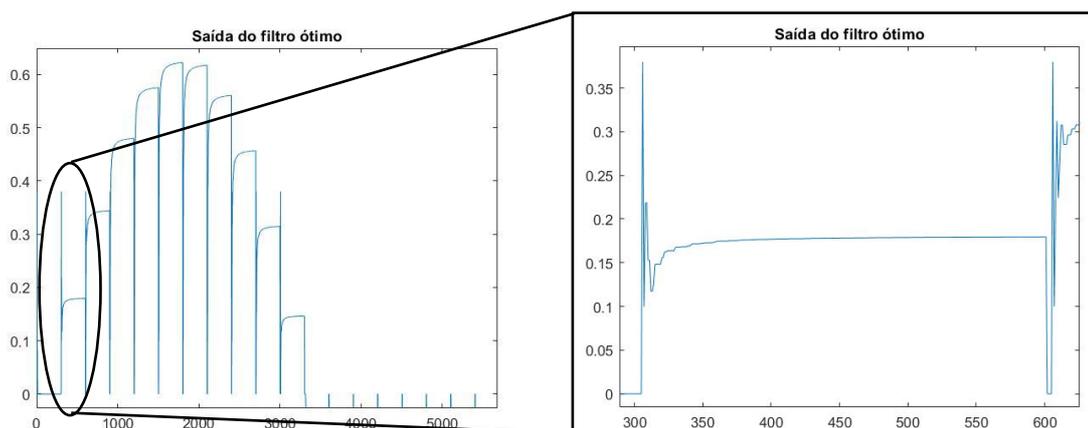


Figura 4.9 - Gráfico referente ao sinal de saída do filtro ótimo

Olhando para o sinal de saída percebe-se que vão sendo calculado limites máximos e mínimos até chegar ao valor final a ser amostrado a uma taxa de OSR, sendo essa a origem das linhas que aparecem superiores a  $V_{in}$ . De notar que para valores mais elevados de  $V_{in}$  o algoritmo consegue obter o valor final afastando-se menos do valor de  $V_{in}$ , quando comparado com o filtro ótimo do modulador de primeira ordem.

### 4.3. Simulações de alto nível

O ambiente de simulação utilizado para as simulações de segunda ordem foi implementado utilizando os mesmos métodos referidos anteriormente para o modulador de primeira ordem. Toda a descrição está feita no capítulo 3.3.1.

As arquiteturas que foram referidas anteriormente são estudadas em alto nível no *Software Octave* de modo a poder observar algumas características. O Modulador implementado em ambas as arquiteturas, é semelhante, portanto a diferença existente entre a performance das duas arquiteturas deve-se ao filtro digital implementado na saída do modulador. Os dois filtros implementados na sequência do modulador são o filtro Col e o filtro ótimo, sendo que foram feitas simulações tendo em conta a presença integradores ideais e também integradores reais, no modulador. Para a implementação do modulador com o filtro Col foi utilizada a arquitetura da Figura 4.10, como filtro ótimo foi utilizada a arquitetura da Figura 4.8.

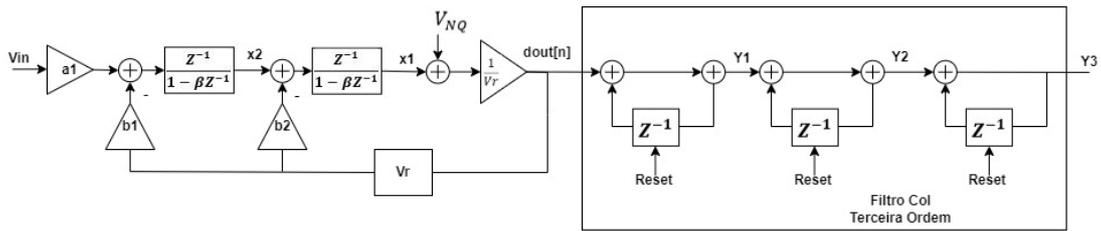


Figura 4.10 - Modulador Sigma-Delta e filtro Col de terceira ordem

O sinal  $V_{in}$ , o sinal de entrada, baseia-se numa senoide de frequência  $\frac{f_{signal}}{F_s}$ , em que  $F_s$  é a frequência de amostragem. Tendo em conta o filtro ótimo, a amplitude de -4 dB, otimiza o SNDR do sistema e para a arquitetura com o filtro Col o valor que maximiza SNDR é 4 dB. Estes valores podem ser confirmados no gráfico da Figura 4.11, onde se variou a amplitude de entrada entre -10 dB e 0 dB e utilizou-se um OSR de 300.

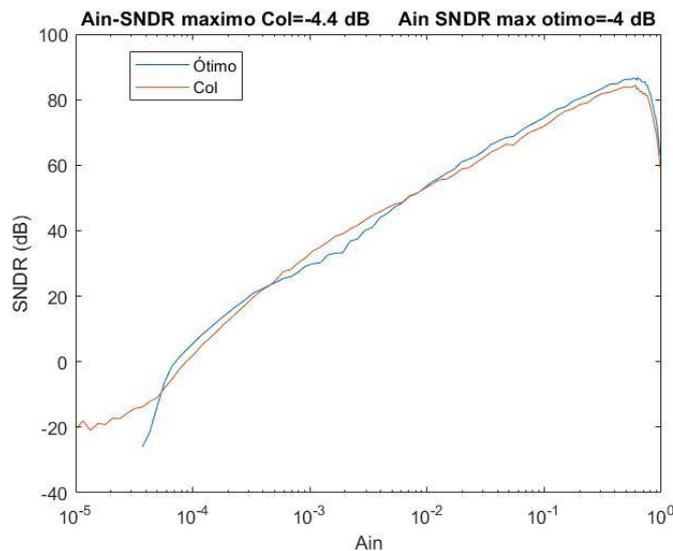


Figura 4.11 - Gráfico de SNDR em função da amplitude de entrada, com ponto máximo de SNDR marcado

Na entrada do modulador é também adicionado o ruído térmico tem como principal causa as vibrações dos condutores de carga que se encontram num condutor elétrico. Nas arquiteturas propostas o ruído térmico é, na sua maioria, proveniente dos *ampops* presentes no modulador e uma pequena parte proveniente dos interruptores. Na Figura 4.12 pode-se observar o sinal  $V_{in}$  em função do tempo, estando assinalados os pontos amostrados e destacado a influência do ruído térmico no sinal de entrada.

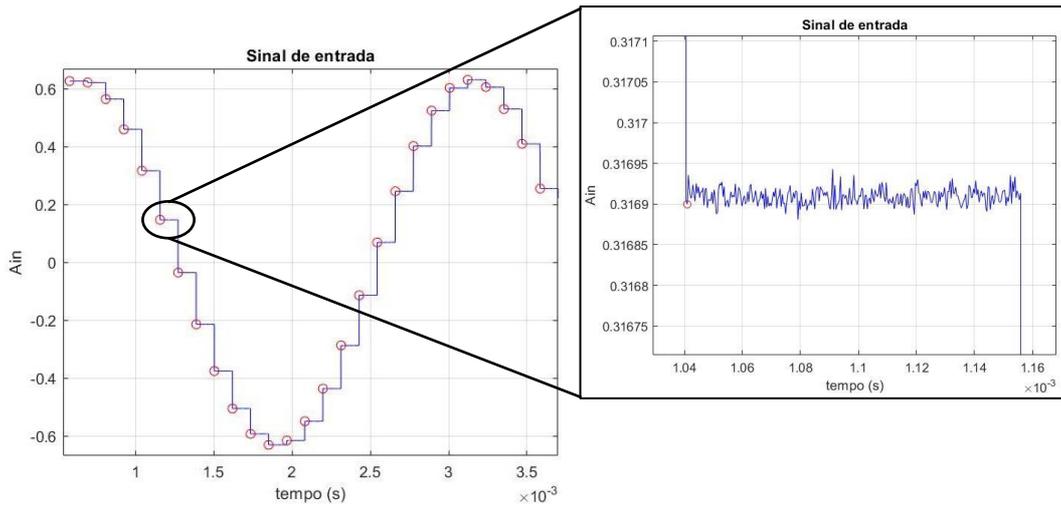


Figura 4.12 Gráfico do sinal de entrada com pontos amostrados e com influência do ruído térmico destacado

De modo a validar o sinal de entrada é apresentada na Figura 4.13 a sua FFT. A frequência utilizada para obter a FFT é linear e começa em 0 e termina em  $\frac{F_s}{2}$ . A FFT permite analisar qual a relação sinal ruído e distorção (SNDR) e assim validar a implementação de  $V_{in}$ .

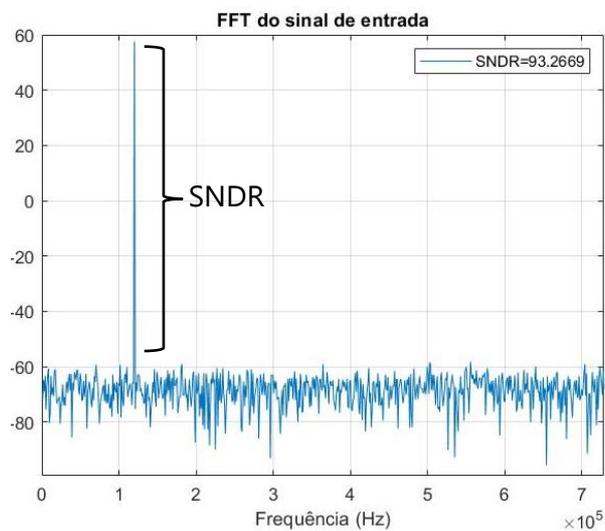


Figura 4.13 - Espectro relativo ao sinal de entrada,  $V_{in}$

As arquiteturas têm alguns parâmetros que podem ser ajustados de modo a tentar obter a melhor SNDR possível. É com base nessa variação que serão estudados os valores máximos que a SNDR pode tomar bem como os valores do  $DR$ . Os valores que irão ser variados ao longo da análise será o coeficiente  $b_2$ , que é multiplicado pela saída

do comparador e pela tensão de referência e subtraído à saída do primeiro integrador, o coeficiente beta que representa a idealidade dos integradores presentes no modulador Sigma-Delta, a taxa de amostragem, OSR, e também o ruído térmico que está presente no sinal  $V_{in}$ . O objetivo da análise passa por encontrar os valores que otimizem tanto a SNDR como o DR.

Inicialmente começou-se por analisar a SNDR nas saídas de cada integrador do filtro Col de terceira ordem, esta análise foi feita com o valor  $beta = 1$ , ou seja, tendo em conta integradores ideais, e arbitrou-se um valor de  $beta=0,99$  para realizar a simulação com integradores reais.

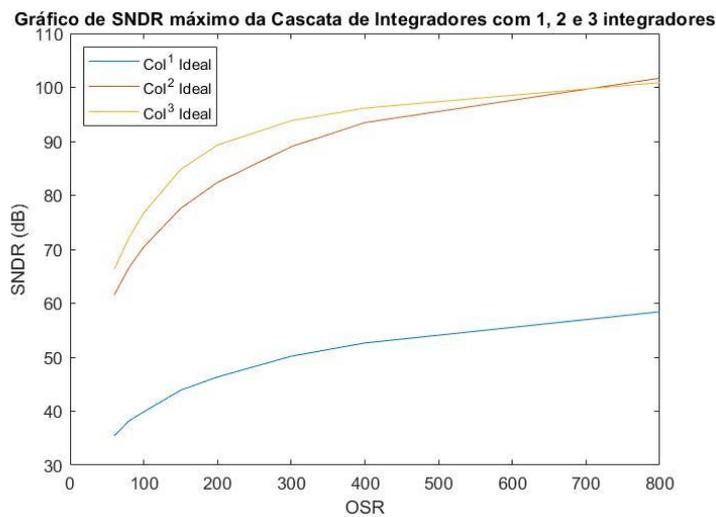


Figura 4.14 - Gráfico de SNDR para as saídas do filtro Col com integradores ideais;  $beta=1$

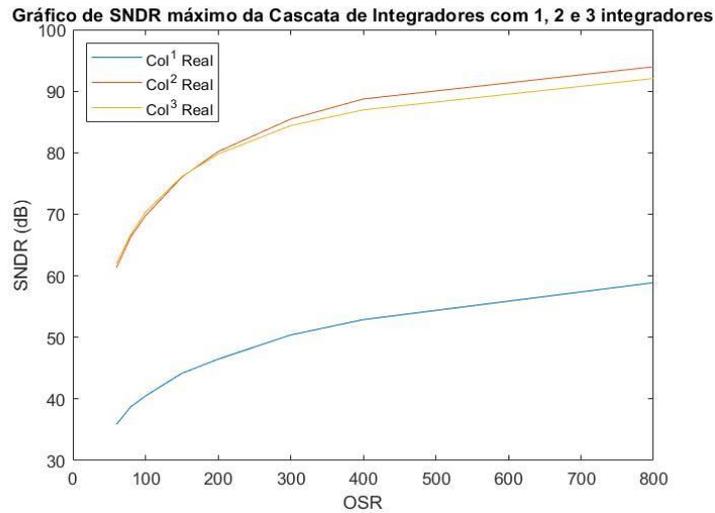


Figura 4.15 - Gráfico de SNDR para as saídas do filtro Col com integradores reais,  $\beta=0,99$

Como resultado dessa simulação resulta o gráfico da Figura 4.14. Analisando é possível entender que tendo em conta apenas os Integradores ideais a utilização de 3 blocos somadores no filtro Col é algo que faz sentido se considerássemos apenas os integradores ideais. Contudo, se observarmos o gráfico Figura 4.15, onde foi aplicado  $\beta = 0,99$ , o valor de SNDR para os vários valores de OSR é superior na saída do segundo integrador do filtro, Y2, comparando com a saída do terceiro integrador, Y3. Sabendo, que não é possível obter um integrador ideal, a análise da arquitetura que utiliza a Col será realizada a partir da saída do segundo integrador do filtro, tornando-o assim num filtro Col de segunda ordem.

A análise que se segue é realizado em simultâneo para a arquitetura implementada com o filtro Col e para a arquitetura implementado com o filtro com algoritmo ótimo, permitindo assim realizar as comparações necessárias entre os resultados obtidos. O primeiro parâmetro estudado é o valor de OSR como tal é feita a variação de OSR entre 60 e 800 e é analisado o comportamento do modulador implementado com o filtro ótimo utilizando integradores reais e ideais e também implementado com o filtro Col de terceira ordem utilizando integradores ideais no modulador e com o filtro Col de segunda ordem quando utilizados integradores reais, isto é, com  $\beta = 0,99$ .

Gráfico do SNDR máximo da Cascata de Integradores e do algoritmo Ótimo(Real e Ideal)

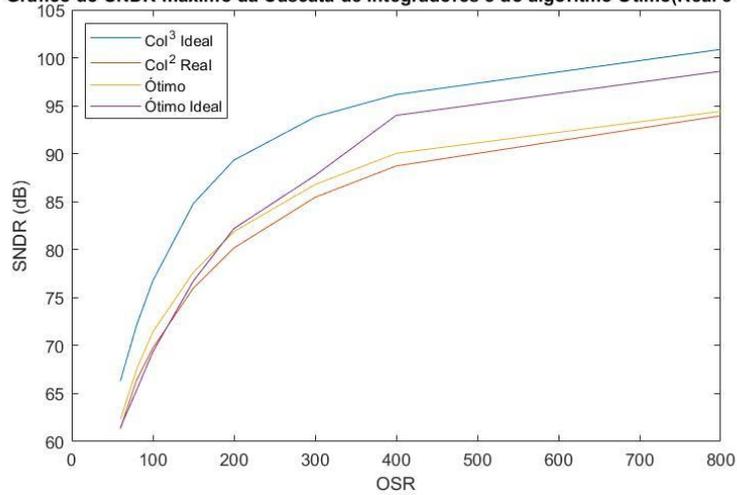


Figura 4.16 - Gráfico do SNDR para a variação de OSR

Gráfico do DR da Cascata de Integradores e do algoritmo Ótimo (Real e Ideal)

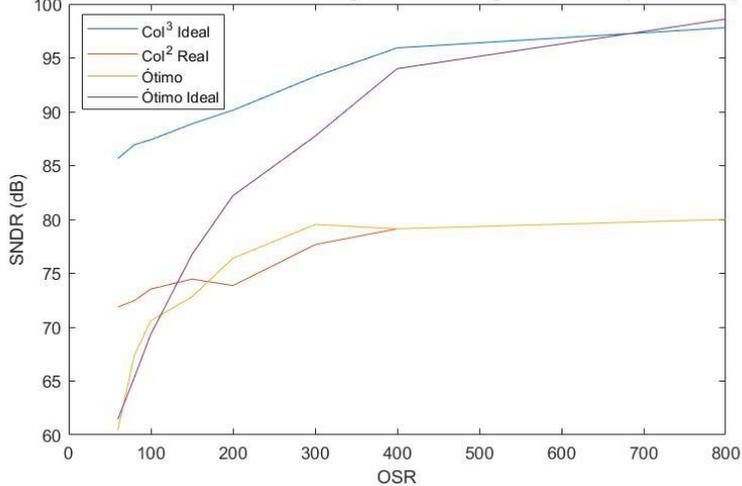


Figura 4.17 - Gráfico de DR para a variação de OSR

Na Figura 4.16 é possível verificar que para integradores ideais a Col permite obter valores mais elevados de SNDR tal como na Figura 4.17 para  $\beta = 1$  o filtro Col consegue melhor desempenho. Se olharmos aos integradores reais é possível observar que a implementação com o filtro ótimo sobressai em relação ao filtro Col, tal como o DR é mais elevado até ao  $OSR = 400$ . Não sendo aconselhável utilizar valores de OSR muito elevados pois diminui bastante a velocidade do conversor, um valor de  $OSR = 300$  permitiria obter uma boa performance. Para este valor de OSR é possível garantir um SNDR 1.5dB mais elevado com o filtro ótimo comparativamente ao filtro Col e DR de mais 2 dB.

O próximo parâmetro a ser estudado será o ruído térmico,  $V_{nt}$ , este parâmetro, introduzido à entrada do modulador foi variado entre  $0\mu V$  e  $500\mu V$ , de modo a perceber como o DR e o SNDR acompanham essa mesma variação. Este ruído provoca pequenas variações no sinal de entrada que dificultam a conversão do sinal provocando algum erro. Na Figura 4.18 e na Figura 4.19 é possível observar o comportamento das métricas estudadas para a variação referida.

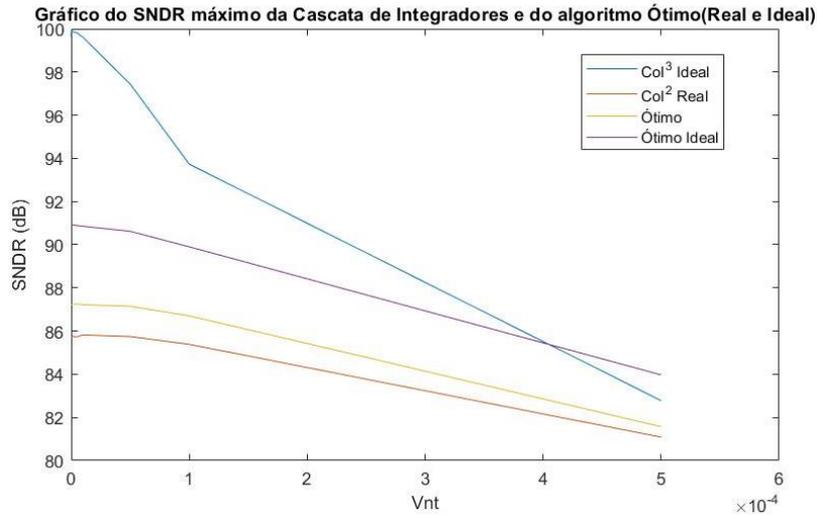


Figura 4.18 - Gráfico do SNDR para a variação de  $V_{nt}$

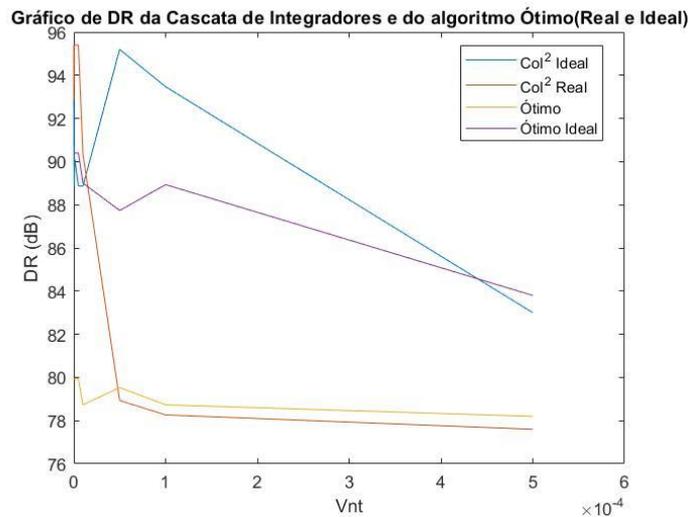


Figura 4.19 - Gráfico do DR para a variação de  $V_{nt}$

Analisando primeiro SNDR, é possível ver que mais uma vez com o modulador com integradores ideais o valor é superior para a Col, cerca de 9 dB para um  $V_{nt} = 10\mu V$ . Tal como aconteceu na situação anterior, para os integradores reais nota-se uma

performance ligeiramente superior na implementação do filtro ótimo, em 1,5dB. Em relação ao DR, tendo em conta os integradores ideais a performance mantém-se num nível aceitável, acima dos 84 dB, inclusive para os valores mais elevados de  $V_{nt}$ . Utilizando os integradores reais destaca-se a melhor performance do Col de segunda ordem que se mantém acima dos 84 dB até cerca de  $V_{nt} = 25\mu V$  e acima do algoritmo ótimo até  $V_{nt} = 50\mu V$ , a partir deste valor de ruído os valores são muito idênticos. De modo a tentar manter um DR relativamente elevado,  $V_{nt} = 25\mu V$  é um valor que é realista em termos práticos e mantém os valores desejados.

De seguida analisou-se a influência do coeficiente b2 que tem influência na integração realizada pelo segundo integrador do modulador. As simulações para este parâmetro foram realizadas variando entre 1.2 e 3. A Figura 4.20 e a Figura 4.21 mostram como b2 influência as métricas SNDR e DR.

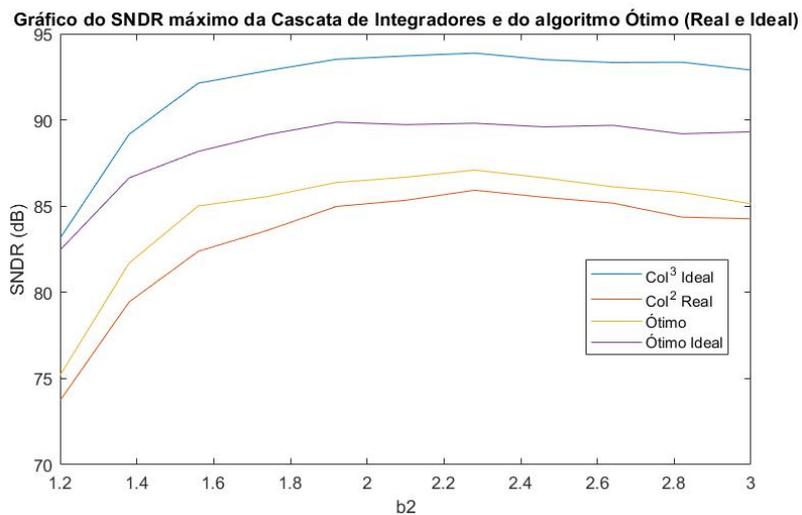


Figura 4.20 - Gráfico do SNDR para a variação coeficiente b2

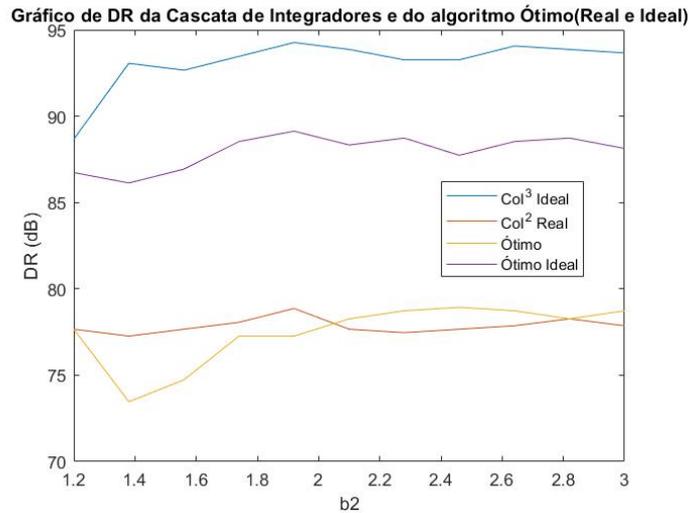


Figura 4.21 - Gráfico do DR para a variação coeficiente b2

Tomando primeiro atenção ao SNDR, tal com já tinha acontecido nas simulações de outros parâmetros o filtro Col com modulador ideal tem um desempenho melhor que o filtro ótimo na mesma situação, por outro lado em situação real destaca-se o filtro ótimo. O valor que permite obter um valor de SNDR mais elevado é  $b2 = 2.28$ , para este valor o filtro ótimo garante mais 1 dB. Em relação a DR assimulações com integradores ideais seguem a mesma lógica, a Col a destacar-se em relação ao algoritmo ótimo. Observando os casos reais, para  $b2 < 2$ , o filtro Col consegue suportar melhor a variação do sinal de entrada, no entanto para valores superiores o filtro ótimo tem 1.5dB a mais de DR.

Por fim o último parâmetro analisado é o valor que introduz a não idealidade no sistema, o  $\beta$ , sendo variado entre 0.90 e 0.995 tem por objetivo perceber como diferentes valores que tornam o sistema não ideal, influenciam tanto o SNDR como o DR.

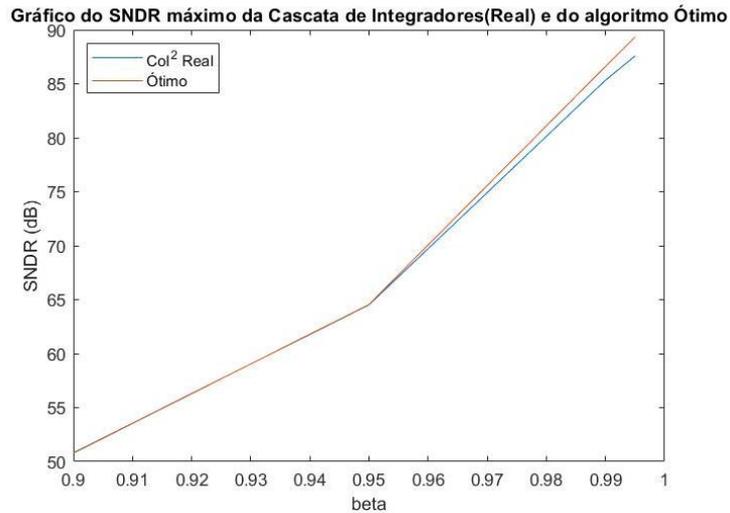


Figura 4.22 - Gráfico do SNDR para a variação do coeficiente beta

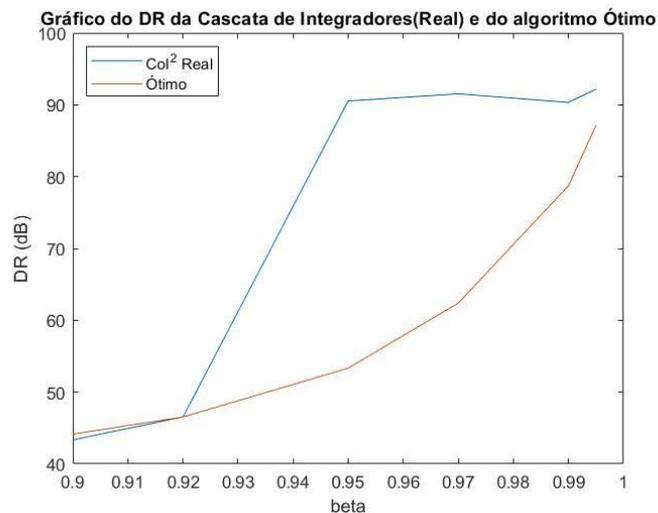


Figura 4.23 - Gráfico do DR para a variação do coeficiente beta

Como se pode ver pela Figura 4.22 ambas as implementações têm um aumento significativo de SNDR com o aumento de beta, no entanto entre ambas a implementação a diferença é praticamente nula. Em relação à Figura 4.23, para valores superiores a 0.92 a Col é claramente superior, sendo que para 0.99 o filtro Col tem mais 5 dB que o ótimo. Sendo  $\beta = 0.99$  o ponto que proporciona o maior valor de SNDR e DR.

Após realizada a análise às várias simulações que avaliam os parâmetros, foi implementada cada uma das arquiteturas com os valores que foram considerados ao longo da análise foi também necessário definir os valores dos coeficientes  $a_1$  e  $b_2$ . Os valores parametrizados estão identificados na Tabela 4.1

Tabela 4.1 - Tabela dos parâmetros obtidos através das simulações para modulador de segunda ordem

	a1	b1	b2	Vnt	Beta	OSR
Real	1	1	2,28	25 $\mu$ V	0,99	300
Ideal					1	

Utilizando os valores dimensionados, sendo que  $\beta = 0.99$  no caso real e  $\beta = 1$  no caso ideal, é possível realizar a FFT tendo em vista a analisar o valor de SNDR e assim comparar os resultados obtidos.

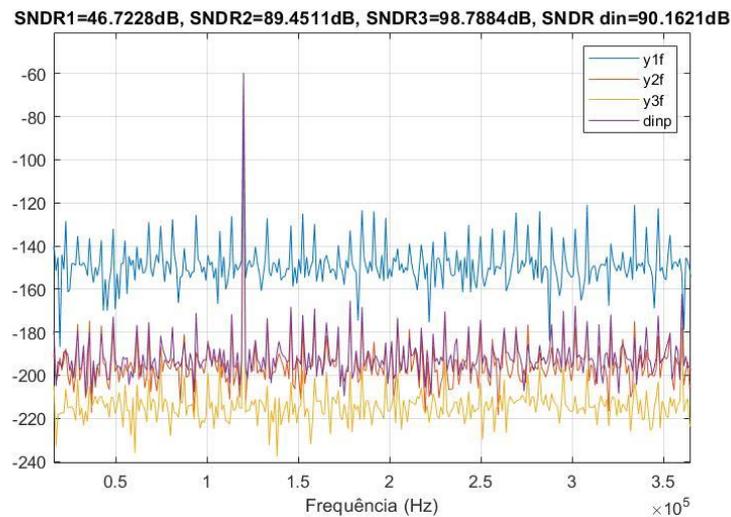


Figura 4.24 - FFT dos sinais de saída com integradores ideais,  $\beta=1$

Como se pode ver na Figura 4.24, que corresponde ao caso ideal analisado, após as primeiras médias a serem amostradas no primeiro integrador o SNDR é 46,7228dB, quando as médias são realizadas uma segunda vez o valor do SNDR passa a 89,4511dB e realizando a terceira média o valor de SNDR alcança os 98,7784dB. Ou seja, no caso dos integradores ideais, é bastante útil utilizar o terceiro somador e realizar a terceira média visto que se consegue obter mais 9dB do que se fosse utilizado um filtro Col de segunda ordem. Em relação à saída do filtro ótimo ideal (dinp), pelo estudo realizado este algoritmo obtém um SNDR semelhante a um filtro Col de segunda ordem.

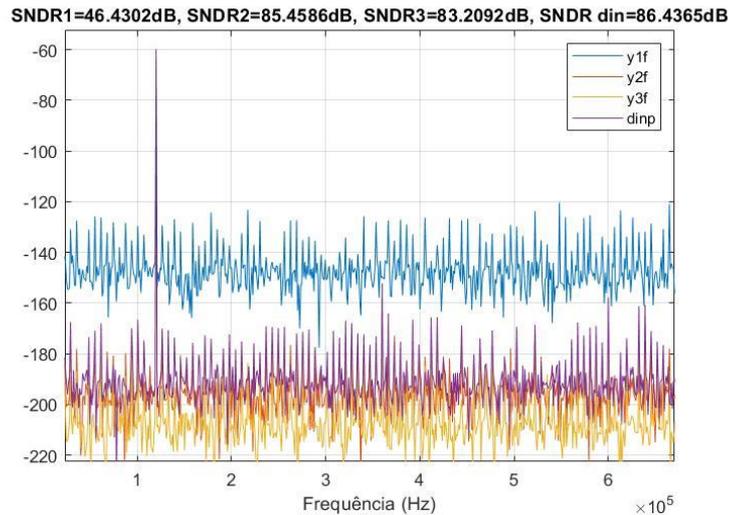


Figura 4.25 - FFT dos sinais de saída com integradores reais,  $\beta=0,99$

A Figura 4.25 corresponde à FFT da arquitetura que contém os integradores reais, ou seja,  $\beta = 0,99$ . Nesta simulação é possível comprovar o que foi referido no início da análise, o SNDR da saída do segundo integrador, ou seja, da segunda média é superior ao SNDR da saída da terceira média o que significa que não há qualquer tipo de benefício em introduzir o terceiro somador, pois não traz nenhum aumento na performance e iria simplesmente ocupar mais área e ser necessário mais hardware. Em relação à arquitetura implementada com o algoritmo ótimo comparando com o obtido na saída do segundo integrador do Col tem mais 1 dB de SNDR, o que representa apenas um ligeiro aumento.

Os valores obtidos tanto para uma arquitetura foram dentro do esperado, principalmente a implementação com filtro Col onde os resultados são bastante positivos, no entanto, comparando com outros autores esperava-se um resultado ligeiramente superior do filtro ótimo.



## CONCLUSÃO E TRABALHO FUTURO

### 5.1. Conclusão

O trabalho realizado permite ter uma ideia mais clara a cerca dos ADC Sigma-Delta e em específico dos ADC Sigma-Delta Incrementais. Ter um conhecimento mais alargado a cerca deste tipo de conversor pode ser determinante para áreas como biomedicina ou de telecomunicações, pois tendo uma noção clara do seu comportamento, é mais fácil definir os seus parâmetros e a forma como otimizar.

Do trabalho realizado foi possível retirar algumas conclusões para cada um dos estudos. Tendo em conta os circuitos para o modulador de 1ª ordem, em termos de integradores ideais o SNDR máximo é claramente superior com o algoritmo ótimo comparando com o resultado para o filtro Col, no entanto introduzindo integradores reais observa-se que o SNDR máximo é apenas 1,5dB superior ao do filtro Col, representando apenas um ligeiro aumento. O que significa que a nível de aplicabilidade real o filtro ótimo apenas terá uma ligeira vantagem, pois não é possível construir um circuito que se comporte como um integrador ideal, porque requereria um amplificador com ganho infinito.

Em relação aos circuitos relacionados com o modulador de 2ª ordem, em primeiro lugar na implementação do filtro Col conclui-se que a implementação de um número superior de médias calculadas, permite um melhor desempenho do ponto de vista ideal. Contudo, do ponto de vista real o mesmo não se verifica, visto que com integradores reais o valor de SNDR máximo é cerca de 2 dB superior para uma Col composta por dois integradores quando comparado com o resultado para três integradores. Portanto, para uma implementação elétrica e conseqüente aplicação ao "mundo real" será vantajoso utilizar apenas 2 integradores no filtro. Realizando a comparação do filtro Col com o algoritmo ótimo, utilizando integradores ideais a implementação do filtro Col de terceira ordem

seria o mais vantajoso tendo mais 8 dB de SNDR máximo que o algoritmo ótimo e mais 9 dB que o filtro Col de segunda ordem. A mesma comparação para os integradores reais permite verificar que o melhor desempenho seria obtido com o algoritmo ótimo, mas apenas com mais 1 dB de SNDR.

Concluindo, apesar de não ter sido possível obter a otimização desejada com o filtro ótimo, esta dissertação pode ser um auxílio para possíveis trabalhos futuros, através dos estudos e análises realizadas.

## 5.2. Trabalho futuro

Para trabalho futuro seria interessante a implementação elétrica, implementando todos os blocos dos vários circuitos em simulações de baixo nível de modo a ser possível analisar como variam os valores dos vários blocos apresentados com a variação dos diferentes componentes elétricos que estão inseridos nos mesmos.

Assim, seria possível enriquecer a análise efetuada nesta dissertação e possivelmente encontrar formas de otimizar e caracterizar melhor os circuitos que foram analisados.

## REFERÊNCIAS

- [1] "Specifications Explained: Spurious-Free Dynamic Range (SFDR) - National Instruments." [Online]. Available: <https://www.ni.com/pt-pt/support/documentation/supplemental/18/specifications-explained--spurious-free-dynamic-range--sfd.html>. [Accessed: 05-Feb-2020].
- [2] W. Kester, "Understand SINAD, ENOB, SNR, THD, THD+ N, and SFDR so You Don't Get Lost in the Noise Floor," *Analog Devices Tutorial, MT-003, Rev. A*, pp. 2–9, 2009.
- [3] "Understanding Delta-Sigma Data Converters - Shanthi Pavan, Richard Schreier, Gabor C. Temes - Google Livros." [Online]. Available: <https://books.google.pt/books?id=JBauDQAAQBAJ&pg=PA20&lpg=PA20&dq=fom+adc&source=bl&ots=hEzdkwJNjx&sig=ACfU3U035Qx9Gq3wKga3v4FidnEMufFmAQ&hl=pt-PT&sa=X&ved=2ahUKEwixzsT5nofnAhXSxoUKHe1uCLIQ6AEwEXoECAgQAQ#v=onepage&q=fom+adc&f=false>. [Accessed: 05-Feb-2020].
- [4] M. P. Ajanya and G. T. Varghese, "Low Power Wallace Tree Encoder for Flash ADC," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 396, no. 1, 2018, doi: 10.1088/1757-899X/396/1/012042.
- [5] V. Kledrowetz and J. Haze, "Basic block of pipelined ADC design requirements," *Radioengineering*, vol. 20, no. 1, pp. 234–238, 2011.
- [6] "Guide to Understanding Successive Approximation Registers (SAR) and Flash ADCs | Maxim Integrated." [Online]. Available: <https://www.maximintegrated.com/en/design/technical-documents/tutorials/1/1080.html>. [Accessed: 05-Feb-2020].
- [7] Y. Lai, "ADC Architectures IV: Sigma-Delta ADC Advanced Concepts and Applications," *Imid 2009*, pp. 1069–1072, 2009.
- [8] R. S. A. Kumar, D. Behera, and N. Krishnapura, "Reset-Free Memoryless Delta-Sigma Analog-to-Digital Conversion," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 65, no. 11, pp. 3651–3661, 2018, doi: 10.1109/TCSI.2018.2854707.
- [9] J. M. De La Rosa, "Sigma-delta modulators: Tutorial overview, design guide, and state-of-the-art survey," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 58, no. 1, pp. 1–21, 2011, doi: 10.1109/TCSI.2010.2097652.
- [10] S. Pavan, N. Krishnapura, R. Pandarinathan, and P. Sankar, "A 90/ $\mu$ W 15-bit  $\Delta\Sigma$  ADC for digital audio," *ESSCIRC 2007 - Proc. 33rd Eur. Solid-State Circuits Conf.*, pp. 198–201, 2007, doi: 10.1109/ESSCIRC.2007.4430279.
- [11] S. Tao, *Power-Efficient Continuous-Time Incremental Sigma-Delta Analog-to-Digital Converters*. 2015.
- [12] M. Yue, D. Wu, and Z. Wang, "Data compression for image sensor arrays using a

- 15-bit two-step sigma-delta ADC," *IEEE Sens. J.*, vol. 14, no. 9, pp. 2989–2998, 2014, doi: 10.1109/JSEN.2014.2321166.
- [13] S. Tao and A. Rusu, "A comparative design study of continuous-time incremental sigma-delta ADC architectures," *Int. J. Circuit Theory Appl.*, vol. 44, no. 12, pp. 2147–2163, 2016, doi: 10.1002/cta.2218.
- [14] W. Yang, D. Kelly, I. Mehr, M. T. Sayuk, and L. Singer, "A 3-V 340-mW 14-b 75-Msample/s CMOS ADC with 85-dB SFDR at Nyquist input," *IEEE J. Solid-State Circuits*, vol. 36, no. 12, pp. 1931–1936, 2001, doi: 10.1109/4.972143.
- [15] D. Behera and N. Krishnapura, "A 2-channel 1MHz BW, 80.5 dB DR ADC using a DS modulator and zero-ISI filter," *Eur. Solid-State Circuits Conf.*, pp. 415–418, 2014, doi: 10.1109/ESSCIRC.2014.6942110.
- [16] L. Dorrer, F. Kuttner, A. Wiesbauer, A. Di Giandomenico, and T. Hartig, "10-Bit, 3 mW Continuous-Time Sigma-Delta ADC for UMTS in a 0.12  $\mu$ m CMOS proces," pp. 245–248.
- [17] K. Nguyen, R. Adams, and K. Sweetland, "A 113dB SNR Oversampling Sigma-Delta DAC for CD/DVD Application," pp. 404–405.
- [18] R. S. A. Kumar and N. Krishnapura, "A 2-Channel ADC Using a Delta-Sigma Modulator Without Reset & a Modulated-Sinc-Sum Filter," *ESSCIRC 2019 - IEEE 45th Eur. Solid State Circuits Conf.*, pp. 365–368, 2019, doi: 10.1109/ESSCIRC.2019.8902610.
- [19] P. Vogelmann, M. Haas, and M. Ortmanns, "A 1.1mW 200kS/s incremental  $\Delta\Sigma$  ADC with a DR of 91.5dB using integrator slicing for dynamic power reduction," *Dig. Tech. Pap. - IEEE Int. Solid-State Circuits Conf.*, vol. 61, pp. 236–238, 2018, doi: 10.1109/ISSCC.2018.8310271.
- [20] V. Quiquempoix *et al.*, "A low-power 22-bit incremental ADC," *IEEE J. Solid-State Circuits*, vol. 41, no. 7, pp. 1562–1571, 2006, doi: 10.1109/JSSC.2006.873891.
- [21] V. Quiquempoix and P. Deval, "US6909388B1 - Fractal sequencing schemes for offset cancellation in sampled data acquisition systems - Google Patents." [Online]. Available: <https://patents.google.com/patent/US6909388B1/en>. [Accessed: 05-Feb-2020].
- [22] V. Quiquempoix, G. Bellini, and J. Collings, "US6788233B1 - Digital decimation filter - Google Patents." [Online]. Available: <https://patents.google.com/patent/US6788233B1/en>. [Accessed: 05-Feb-2020].
- [23] S. Tao and A. Rusu, "A power-efficient continuous-time incremental sigma-delta ADC for neural recording systems," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 62, no. 6, pp. 1489–1498, 2015, doi: 10.1109/TCSI.2015.2418892.
- [24] K. Singh and S. Pavan, "A 14 Bit Dual Channel Incremental Continuous-Time Delta Sigma Modulator for Multiplexed Data Acquisition," *Proc. IEEE Int. Conf. VLSI Des.*, vol. 2016-March, pp. 230–235, 2016, doi: 10.1109/VLSID.2016.21.
- [25] A. Agah, K. Vleugels, P. B. Griffin, M. Ronaghi, J. D. Plummer, and B. A. Wooley, "A high-resolution low-power incremental  $\Sigma$ ADC with extended range for biosensor arrays," *IEEE J. Solid-State Circuits*, vol. 45, no. 6, pp. 1099–1110, 2010, doi: 10.1109/JSSC.2010.2048493.
- [26] P. Vogelmann, J. Wagner, A. Hänslér, and M. Ortmanns, "Improved SQNR and MSA in Incremental  $\Delta\Sigma$  Modulators by Using a Recuperation Phase," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 65, no. 5, pp. 557–561, 2018, doi: 10.1109/TCSII.2018.2821365.

- [27] B. Wang, M. K. Law, S. B. Belhaouari, and A. Bermak, "Near-Optimal Decoding of Incremental Delta-Sigma ADC Output," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 67, no. 11, pp. 3670–3680, 2020, doi: 10.1109/TCSI.2020.3010991.
- [28] S. Kavusi, H. Kakavand, and A. El Gamal, "On Incremental Sigma – Delta Modulation," vol. 53, no. 5, pp. 1004–1015, 2006.
- [29] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, 1st ed. Springer, Boston, MA, 1992.

## Anexo A - Código simulações do modulador de segunda ordem

```
%simulação de um modulador de segunda ordem
close all
clear all
%definições
%definições do modulador
Vr=1;

b1=1;
% Vnt=[0 0.1e-6 0.5e-6 1e-6 5e-6 10e-6 50e-6 100e-6];
Vnt=10e-6;
% beta=[0.9, 0.95, 0.99, 0.995];
beta=0.99;%0.99; %Não ideal
% osr=[400 800 1000 1100 1200 1500 2000]; %osr(i)=60
osr=[1200];

% osr_s=10; %OSR sinal inteiro
Nmedias=1200;
%Nmedias(i)1=300;
B=20e3;
M=0;
%%
% Ain=0.3;
np_ain=100;

Aindb(1)=-4;

Ain(1)=10^(Aindb(1)/20)*Vr;

p=111;

osr_s=1200/p; %OSR sinal fracionário

% Variação do Ain
for k=2:100
    if Aindb(k-1)<mag2db(Vr)-6
        Aindb(k)=Aindb(k-1)+95/(np_ain*3/4);
    else
        Aindb(k)=Aindb(k-1)+5/(np_ain*1/4);
    end
    Ain(k)=10^(Aindb(k)/20)*Vr;
end
y1=zeros(1,p*10);
y2=zeros(1,p*10);
y3=zeros(1,p*10);
sndr_y1d=zeros(length(Ain),1);
sndr_y2d=zeros(length(Ain),1);
sndr_dind=zeros(length(Ain),1);
```

```

for g=1:length(osr)
%   for i=1:length(beta)
    for k=1:length(Ain)
        %sinal de 9
        fsinal=0.4e3;

        Fs=2*osr*osr_s*fsinal*1;
        extra=5; %ciclos extra para a resposta impulsiva
        np=round(p*Fs/fsinal+0+extra*Nmedias);
        nn=(0:np)-mod(0:np,(osr)); %Com S&H
        %nn=(0:np); %Sem S&H
        vin=sin(2*pi*fsinal*nn/Fs)*Ain(k)+randn(1,np+1)*Vnt(g);
        time=0:1/Fs:np/Fs;
        lbmax=-1;
        ubmin=1;
        Sn=zeros(1,np);
        plb=1;
        pub=1;
        din=zeros(1,np);
        x2=zeros(1,np);
        dout=zeros(1,np);
        nnn=0;

        for n=2:(np+1)
            if mod(n-3,Nmedias)==0
                x2(n-1)=0; %reset integradores modulador
                y1(n-1)=0; %reset integradores filtro
                y2(n-1)=0; %reset integradores filtro
                Sn(n-1)=0;
                ubmin=1;
                lbmax=-1;
                din(n-1)=0;
            end
            dout(n)=sign(x2(n-1));
            x2(n)=vin(n)-b1*Vr*dout(n)+beta*x2(n-1); %utilizar o sinal amo-
trado (valor constante durante Nmedias(i))

            %filtro ótimo
            %%%%%%%%%
            NN=mod(n-3,Nmedias);
            din(n)=din(n-1);
            if dout(n)<0
                if Sn(n-1)/NN<=ubmin
                    ubmin=Sn(n-1)/NN;
                    pub=NN;
                    din(n)=ubmin;
                end
            else
                if Sn(n-1)/NN>=lbmax
                    lbmax=Sn(n-1)/NN;
                    plb=NN;
                    din(n)=lbmax;
                end
            end
        end
    end
end

```

```

        end
    end
    Sn(n)=Sn(n-1)+dout(n);

    if plb>pub
        din(end)=lbmax;
    else
        din(end)=ubmin;
    end

    y1(n)=y1(n-1)+dout(n)/Nmedias;
    y2(n)=y2(n-1)+y1(n)/Nmedias;
%
end

y1d=y1(1:Nmedias:np);
y2d=y2(1:Nmedias:np);
y3d=y3(1:Nmedias:np);

%ver saida
[num,den]=butter(4,1/(200));
%[num,den]=cheby1(6,0.2,1/(2.1*osr(i)))
outf=filter(num,den,dout);
%
%
janela=ones(1,round((np-extra*Nmedias-2)/Nmedias));
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
y1ff=fft(y1((extra+1)*Nmedias+1:osr:end).*janela);
y1p=y1ff.*conj(y1ff)/np^2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
y2ff=fft(y2((extra+1)*Nmedias+1:osr:end).*janela);
y2p=y2ff.*conj(y2ff)/np^2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dinff=fft(din((extra+1)*Nmedias+1:osr:end).*janela);
dinp=dinff.*conj(dinff)/np^2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Psd=sum(y1p(p+1-1:p+1+1))
PNd=sum(y1p(1:end/2))-Psd
sndr_y1d(k,g)=10*log10(Psd/PNd);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Psd=sum(y2p(p+1-1:p+1+1))
PNd=sum(y2p(1:end/2))-Psd
sndr_y2d(k,g)=10*log10(Psd/PNd);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Psd=sum(dinp(p+1-1:p+1+1))
PNd=sum(dinp(1:end/2))-Psd
sndr_dind(k,g)=10*log10(Psd/PNd); %trocar i por g para o OSR

```

```

end
snдр_maxy1(1,g)=max(snдр_y1d(:,g));           %valor maximo snдр
snдр_maxy2(1,g)=max(snдр_y2d(:,g));           %valor maximo snдр
snдр_maxdind(1,g)=max(snдр_dind(:,g));         %valor maximo snдр
Hy1=min(abs(snдр_y1d(:,g)));                 %Ain para snдр=0
Hy2=min(abs(snдр_y2d(:,g)));                 %Ain para snдр=0
Hdind=min(abs(snдр_dind(:,g)));              %Ain para snдр=0
Ain_snдр_zeroy1=Ain(find(abs(snдр_y1d(:,g))==Hy1,1));
Ain_snдр_zeroy2=Ain(find(abs(snдр_y2d(:,g))==Hy2,1));
Ain_snдр_zerodind=Ain(find(abs(snдр_dind(:,g))==Hdind,1));
Ain_snдр_maxy1=(Ain(snдр_y1d(:,g))==snдр_maxy1(1,g));           %Ain para
snдр MAX
Ain_snдр_maxy2=(Ain(snдр_y2d(:,g))==snдр_maxy2(1,g));           %Ain para
snдр MAX
Ain_snдр_maxdind=Ain(snдр_dind(:,g))==snдр_maxdind(1,g));       %Ain
para snдр MAX
DR_dBy1(1,g)=20*log10(Ain_snдр_maxy1/Ain_snдр_zeroy1);
DR_dBy2(1,g)=20*log10(Ain_snдр_maxy2/Ain_snдр_zeroy2);
DR_dBdind(1,g)=20*log10(Ain_snдр_maxdind/Ain_snдр_zerodind);
end

```

## Anexo B - Código simulações do modulador de primeira ordem

```
%simulação de um modulador de segunda ordem
close all
clear all
%definições
%definições do modulador
Vr=1;
b2=2.28;%1.8;
% b2=1.2:1.8/10:3;
b1=1;
% Vnt=[0 0.1e-6 0.5e-6 1e-6 5e-6 10e-6 50e-6 100e-6 500e-6];
Vnt=10e-6;%10e-6provavelmente
% beta=[0.9, 0.92, 0.95, 0.97, 0.99, 0.995];
beta=1; %Não ideal

osr=[60 80 100 150 200 300 400 800]; %osr(g) (i)=60
% osr=[300];

Nmedias=osr;
%Nmedias(g) (i)1=300;
B=20e3;

%%%
% Ain=0.3;
np_ain=100;

Aindb(1)=-100;

Ain(1)=10^(Aindb/20)*Vr;

%osr_s=10; %OSR sinal inteiro
p=111;
osr_s=1200/p; %OSR sinal fracionário

%Variação do Ain
for k=2:100
    if Aindb(k-1)<mag2db(Vr)-6
        Aindb(k)=Aindb(k-1)+95/(np_ain*3/4);
    else
        Aindb(k)=Aindb(k-1)+5/(np_ain*1/4);
    end
    Ain(k)=10^(Aindb(k)/20)*Vr;
end
y1=zeros(1,p*10);
y2=zeros(1,p*10);
y3=zeros(1,p*10);
```

```

sndr_y1d=zeros(length(Ain),1);
sndr_y2d=zeros(length(Ain),1);
sndr_y3d=zeros(length(Ain),1);
sndr_dind=zeros(length(Ain),1);

for g=1:length(osr)
%   for i=1:length(beta)
%       for k=1:length(Ain)
%           %sinal de entrada
fsinal=0.4e3;
Fs=2*osr(g)*osr_s*fsinal*1;
extra=5; %ciclos extra para a resposta impulsiva
np=p*Fs/fsinal+0+extra*Nmedias(g);
nn=(0:np)-mod(0:np,osr(g)); %Com S&H
%nn=(0:np); %Sem S&H
vin=sin(2*pi*fsinal*nn/Fs)*Ain(k)+randn(1,np+1)*Vnt;
media_vin(g,k)=sum(vin)/length(vin);
%   vin=-1+2/(p*10)*(0:p*10-1);
%   %vin=sign(vin)*Ain;
time=0:1/Fs:np/Fs;
%
lbmax=-1;
ubmin=1;
Sn=0;
plb=1;
pub=1;
intel=0;
a3=0;
Rn=a3;
a1=1;
a2=0;
c1=0;
din=zeros(1,np);%*10);
%
x1=zeros(1,np);
x1(1)=-1e-12;
x2=zeros(1,np);
dout=zeros(1,np);
nnn=0;

for n=2:(np+1)
    if mod(n-3,Nmedias(g))==0
        x2(n-1)=0; %reset integradores modulador
        x1(n-1)=0; %reset integradores modulador
        y1(n-1)=0; %reset integradores filtro
        y2(n-1)=0; %reset integradores filtro
        y3(n-1)=0; %reset integradores filtro
        Sn=0;
        Rn=0;
        ubmin=1;
        lbmax=-1;
        intel=0;
    end
end

```

```

        din(n-1)=0;
        nnn=0;
    end
    dout(n)=sign(x1(n-1));
%   vnin(n)=vnin(n)*10^(10/20); %para aumentar 10 dB de ruído
    x2(n)=vin(n)-b1*Vr*dout(n)+beta*x2(n-1); %utilizar o sinal amos-
trado (valor constante durante Nmedias(g)(i))
    x1(n)=x2(n-1)-b2*Vr*dout(n)+beta*x1(n-1);

%filtro ótimo
%*****
    din(n)=din(n-1);
    nnn=nnn+1;
    Rn=Rn+a1*(nnn-1)+(a1*c1+a2);
    if dout(n)<0
        if Sn/Rn<=ubmin
            ubmin=Sn/Rn;
            pub=n;
            din(n)=ubmin;
        end
    else
        if Sn/Rn>=lbmax
            lbmax=Sn/Rn;
            plb=n;
            din(n)=lbmax;
        end
    end
    if n>=M
        Sn=Sn+(b1*c1+b2)*dout(n)+b1*intel;
        intel=intel+dout(n);
    end
    if plb>pub
        din(end)=lbmax;
    else
        din(end)=ubmin;
    end
end

y1(n)=y1(n-1)+dout(n)/Nmedias(g);

y2(n)=y2(n-1)+y1(n)/Nmedias(g);

y3(n)=y3(n-1)+y2(n)/Nmedias(g);
end

y1d=y1(1:osr(g):np);
y2d=y2(1:osr(g):np);
y3d=y3(1:osr(g):np);

%   %ver saida
[num,den]=butter(4,1/(200));
%[num,den]=cheby1(6,0.2,1/(2.1*osr(g)(i)))
outf=filter(num,den,dout);

```

```

% %          %%%%%%%%%%FFT_y1%%%%%%%%%
janela=ones(1,(np-extra*Nmedias(g)-0)/Nmedias(g));
y1ff=fft(y1((extra+1)*Nmedias(g)+1:osr(g):end).*janela);
y1p=y1ff.*conj(y1ff)/np^2;

%%%%%%%%%FFT_y2%%%%%%%%%
y2ff=fft(y2((extra+1)*Nmedias(g)+1:osr(g):end).*janela);
y2p=y2ff.*conj(y2ff)/np^2;

%%%%%%%%%FFT_y3%%%%%%%%%
y3ff=fft(y3((extra+1)*Nmedias(g)+1:osr(g):end).*janela);
y3p=y3ff.*conj(y3ff)/np^2;

%%%%%%%%%FFT_din%%%%%%%%%
dinff=fft(din((extra+1)*Nmedias(g)+1:osr(g):end).*janela);
dinp=dinff.*conj(dinff)/np^2;

% %
%          %%%%%%%%%%SNDR_y1%%%%%%%%%
Psd=sum(y1p(p+1-1:p+1+1))
PNd=sum(y1p(1:end/2))-Psd
snдр_y1d(k,g)=10*log10(Psd/PNd); %trocar i por g para o OSR
%%%%%%%%%SNDR_y2%%%%%%%%%
Psd=sum(y2p(p+1-1:p+1+1))
PNd=sum(y2p(1:end/2))-Psd
snдр_y2d(k,g)=10*log10(Psd/PNd); %trocar i por g para o OSR
%%%%%%%%%SNDR_y3%%%%%%%%%
Psd=sum(y3p(p+1-1:p+1+1))
PNd=sum(y3p(1:end/2))-Psd
snдр_y3d(k,g)=10*log10(Psd/PNd); %trocar i por g para o OSR
%          %%%%%%%%%%SNDR_DIND%%%%%%%%%
Psd=sum(dinp(p+1-1:p+1+1))
PNd=sum(dinp(1:end/2))-Psd
snдр_dind(k,g)=10*log10(Psd/PNd); %trocar i por g para o OSR

end

snдр_maxy1(1,g)=max(snдр_y1d(:,g));          %valor maximo snдр
snдр_maxy2(1,g)=max(snдр_y2d(:,g));          %valor maximo snдр
snдр_maxy3(1,g)=max(snдр_y3d(:,g));          %valor maximo snдр
snдр_maxdind(1,g)=max(snдр_dind(:,g));          %valor maximo snдр
Hy1=min(abs(snдр_y1d(:,g)));          %Ain para snдр=0
Hy2=min(abs(snдр_y2d(:,g)));          %Ain para snдр=0
Hy3=min(abs(snдр_y3d(:,g)));          %Ain para snдр=0
Hdind=min(abs(snдр_dind(:,g)));          %Ain para snдр=0
Ain_snдр_zeroy1=Ain(find(abs(snдр_y1d(:,g))==Hy1,1));
Ain_snдр_zeroy2=Ain(find(abs(snдр_y2d(:,g))==Hy2,1));
Ain_snдр_zeroy3=Ain(find(abs(snдр_y3d(:,g))==Hy3,1));
Ain_snдр_zerodind=Ain(find(abs(snдр_dind(:,g))==Hdind,1));
Ain_snдр_maxy1=(Ain(snдр_y1d(:,g))==snдр_maxy1(1,g));          %Ain para
snдр MAX
Ain_snдр_maxy2=(Ain(snдр_y2d(:,g))==snдр_maxy2(1,g));          %Ain para
snдр MAX

```

```

    Ain_sndr_maxy3=(Ain(sndr_y3d(:,g)==sndr_maxy3(1,g)));           %Ain para
sndr MAX
    Ain_sndr_maxdind=Ain(sndr_dind(:,g)==sndr_maxdind(1,g));       %Ain para
sndr MAX
    DR_dBy1(1,g)=20*log10(Ain_sndr_maxy1/Ain_sndr_zeroy1);
    DR_dBy2(1,g)=20*log10(Ain_sndr_maxy2/Ain_sndr_zeroy2);
    DR_dBy3(1,g)=20*log10(Ain_sndr_maxy3/Ain_sndr_zeroy3);
    DR_dBdind(1,g)=20*log10(Ain_sndr_maxdind/Ain_sndr_zerodind);
end

```